



Cool Auditing Tool - for COBOL

PC Software for analysis and auditing of COBOL Programs

Version 2.00 (October 1, 2012)

COBOL tools for the 21st Century

Software User Guide

Marble Computer, Inc. (800) 252-1400 info@marblecomputer.com

Printing and Document Information

Copyright © 2010, 2011, 2012 by Marble Computer, Inc.

Release 2.0

Printed in the United States of America

Trademarked names appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of a trademarked name, Marble Computer, Inc. states that it is using the names only for editorial purposes and to the benefit of the trademark owner with no intention of infringing upon that trademark.

Important

With practice and learning, all the functionality of the CAT System, productivity and creativity will be greatly enhanced and you will become comfortable with its operation. We recommend you carefully review all the operational features contained in this user guide.

Customer Support

The CAT Software System has been designed and engineered to be easy to use. Answers to many technical or operational questions you may have can be found in this Official Reference for the Installation and Operation of the CAT System, which also contains the latest list of most commonly asked questions and troubleshooting tips.

Customer support may be obtained by several methods.

- *Refer to this general user guide including the trouble-shooting guide located at the end of this document.*
- For regular technical support for the CAT, contact Marble Computer, Inc. You can call or e-mail as follows:

Phone: (800) 252-1400

Email: tech@marblecomputer.com

History

Marble Computer, Inc. has been in business since 1983 when they purchased DCD III, a mainframe software package for the maintenance of COBOL programs. DCD III was originated in 1975 and we have mainframe clients who have been using DCD III since that time. Cool Auditing Tool (CAT) takes the best features of DCD III for the mainframe and adds a number of new advanced features for PC auditing and analysis of COBOL and also take advantage of the full functionality available to PC applications.

Introduction.....	1
Features provided by the CAT Software System	3
Cool Auditing Tool Window.....	4
Display of Cool Auditing Tool Window	4
Alternate Compile Listing.....	5
Sample Listing with some Data and Procedure Division	6
Cobol Syntax Editor.....	7
Charting	8
Hierarchy Chart	8
Flowchart.....	8
Overview	9
Overview of Cool Auditing Tool for COBOL.....	9
CAT Main Display.....	9
File drop down menu	9
Dialog to Select COBOL program.....	10
Dialog to Select COBOL Copy Members	10
Display on left side after doing Run	11
Display Narrative on right side after Selecting a field	12
Quick View Window doing Forward Tracing.....	13
Quick View Window doing Backward Tracing	14
Perform Warnings.....	14
COBOL Errors	15
Options Tab.....	15
Narrative Selection Tab	16
Narrative Formatting control	16
Alternate Compile Listing button	17
Alternate Compile Listing Source Listing	18
Quick View Window (QVW).....	21
Tabs just above the QVW.....	21
Display of Tab Names	21
QVW Forward Tracing.....	21
Starting at Procedure Division begin.....	22
Adding other COBOL instructions into the QVW	23
Restarting the QVW to start at a different PERFORM.....	24
Enhanced detail within the QVW Display	24
QVW Backward Tracing	25
Using yellow highlighted checkbox to invoke Backward Tracing	25
Tracing back to Program-Begin OR to more than one routine performing a PERFORM	26
Tracing further back from a previously blue highlighted PERFORM	27
After doing a RESET from Tracing further back.....	27
Preparing to restart Forward Tracing from a point within Backward Tracing	28
Forward Tracing restarted from a previous Backward Tracing PERFORM	28
What is Unique to Forward Tracing	29
What is Unique to Backward Tracing.....	29

Perform Warnings - 2nd Tab.....	30
Description of Warnings and Errors	30
COBOL Perform Warnings Window.....	31
Four types of Information.....	32
COBOL Errors - 3 rd Tab	33
COBOL Errors Sample	33
Narrative Display	34
Using the +/- Button.....	34
All Narrative Elements	34
Procedure Division Only	35
COBOL Narrative (two types)	35
DATA DIVISION NARRATIVE	36
PROCEDURE DIVISION NARRATIVE.....	39
Narrative Examples:	40
INDIRECT REFERENCES	44
CAT Navigation Button Operation.....	45
Narrative Navigation Operation	45
Narrative upper control block section.....	45
Narrative +/- toggle options.....	46
Quick View Navigation Operation	47
QVW Navigation Buttons.....	47
Modify Program Settings.....	48
Program Options	48
Modify Report Options	51
Report Options	51
Tab Options Selection	54
Viewing the Lower Tab Section	54
Options - 4 th Tab	54
Perform Warnings Options Detailed Description	56
COBOL Errors Detailed Description.....	57
Highlight Lines in Source Listing Detailed Description	58
Narrative Selection - 5 th Tab	59
Data Narrative / Data Division Options	60
Data Narrative / Procedure Division Options	62
Alternate Compile Listing	64
Introduction to the Alternate Compile Listing Facility	64
Overview of Alternate Compile Listing (ACL) Facility Reports.....	65
Source Listing Report - Procedure Division	67
Alternate Compile Listing Navigation Tools	81
CAT Editor.....	83
Launching the Editor	83
Menu bar	84

Toolbar	92
Project explorer window	93
Errors Window	96
Hierarchy Charts.....	97
Flowcharts.....	100
Special Considerations.....	101
Incorrectly Defined Mainline Routine - Caused by missing Stop Run type verb	101
Definition of Program Begin.....	103

Introduction

Cool Auditing Tool (CAT) provides unique tracing capability for auditing and analysis of COBOL programs.

Some of our features are based on the 35+ year history of DCD III and additional features were added such as our current ability to do instant forward and backward tracing incorporating **both** major functions of analyzing computer programs. These two functions are listed here:

- Follow computer logic back and forth
 - In Cobol Procedure Division Analysis
- Follow the flow of data while following computer logic
 - In Cobol, following the flow of data fields residing in the Data Division while doing Procedure Division Analysis

The main window of CAT actually has 3 windows present at one time as listed here:

- The main COBOL window showing the source listing with COPY members expanded.
- A COBOL Data Division Narrative window, that for user selected fields, shows all Data Division Attributes and all Procedure Division references in a format dictated by the user.
- A third window that has many uses. The main use of this window, however, is a Quick View Window that allows the user to do quick tracing for the flow of computer logic forwards and also backwards. Tracing of selected data fields is also available in the Quick View Window during forward tracing.

Having the Quick View Window available as one of the three significant windows just mentioned allows a COBOL programmer or technician to see relevant transfer of control from program begin through several routines at once or backwards transfer of control from a particular routine back to program begin.

- During forward tracing this optionally includes the use of selected data fields within each routine

- During backward tracing, where there is more than one path backwards, tracing is guided and followed through mouse clicked choice giving a very clear understanding of all of the paths that reach this routine
- Transitioning back and forth within forward or backward tracing or transitioning between forward and backward tracing is done very quickly via mouse clicking.

The end result is very fast tracing through a COBOL program making analysis easier and more complete resulting in great time savings and a lot less errors in maintaining COBOL code.

Another feature of CAT is complete tracing of PERFORM logic to show warnings in Performs and also out and out typically unseen errors in Perform usage.

For programmers familiar with using DCD III and the Alternate Compile Listing within that software, this feature is available, and now on the PC. This report has many features that summarize other features for COBOL users. For instance, with the Alternate Compile Listing is the ability to trace the entire program reference to CALLs and COPY members at one glance. Unused Data Names and dead code are also found in this report. Many other features are also reported here.

CAT can also facilitate analysis using charting tools to graphically represent program structure.

Each program brings in files (or records of data fields) and creates or modifies other fields of data. These files are typically used in other programs in whatever sequence is needed for this COBOL program.

Where CAT (Cool Auditing Tool) and the mainframe software DCD III provide the most support is in the maintenance (auditing and analysis) of COBOL programs, in other words, the internals of the COBOL program.

DCD III has provided assistance in the two functions mentioned above with the largest assistance provided in understanding the flow of Data Fields with the Narrative created for each data field. This eliminates the need for constant visiting of the cross reference map and going back and forth from there to various places in the Procedure Division to understand how each data field is used. More on this Narrative is covered later in this document.

CAT includes a code editor that enables users to modify programs as needed while analyzing their source code. CAT's editor works in a window separate from the main CAT window, allowing the user to analyze a program freely on the one hand, and write or modify it freely on the other hand – without the two functions interfering with each other.

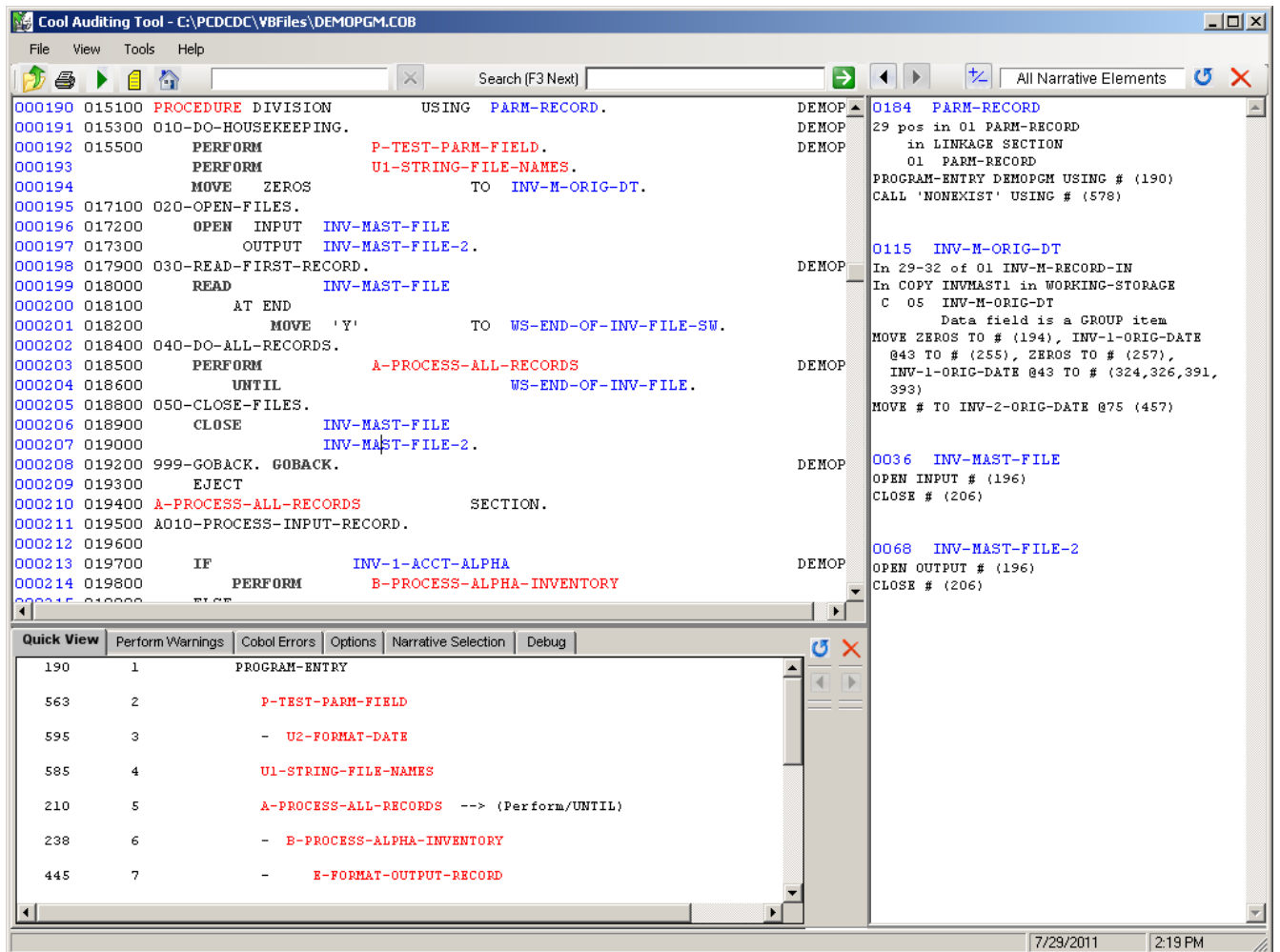
Features provided by the CAT Software System

CAT provides a number of additional features that will ease the burden of analysis by a COBOL programmer. These include:

- Advanced PC Windows graphical user interface
- Mouse click selection on highlighted paragraph and data name to provides instant information
- An intelligent Editor allows writing and modification of code as analysis tools aid in understanding it.
- Other control buttons as well as options that make the selection of information in greater or lesser detail very easy
- Three similar, but slightly different, display formats are provided for the Data Field Narrative including two in the CAT Narrative Display and another in the Quick View window
- Forward tracing of Performed PARAGRAPHS (and/or SECTIONS) are shown in the Quick View Window and can be updated with the single click of the mouse
- Backward Tracing from any Performed outline allows analysis of how any performed routine is referenced all the way back to PROGRAM-ENTRY
- Forward and Backward buttons allow interactive viewing.
- Both following logic and understanding the flow of Data are brought together in the Quick View Window by allowing selected Data Narrative to be listed directly underneath performed PARAGRAPH names.
- Unused code or dead code is shown for Paragraphs and Sections.
- PERFORM Warnings show possible loss of control in a unique format allowing better understanding of COBOL code and identifying possible dangerous errors.
- Errors are available for viewing at a single click.
- Expanded information is provided in the CAT Narrative window to provide formatted and timely information of data fields that are selected.
- Tabs where the Quick View Window resides allow viewing of other reporting information with formatting options available.
- Flowcharts and hierarchy charts visually represent program logic.

Cool Auditing Tool Window

There are three basic windows in the CAT system. These are not to be confused with the available ACL listing which is a totally separate feature. These three windows are designed to be viewed all at once in one oversized window.



Display of Cool Auditing Tool Window

- The source listing with COPY members included is the main and largest window and resides to the left and top of all other windows.
- The Narrative window which shows every piece of information concerning any selected data name is on the right side of the other two windows and all information is formatted to fit on one or more lines with a maximum length of 43 characters from left to right.

- The Quick View Window resides at the bottom and provides, as it's name suggests, a quick view of what happens forward from any place in the Procedure Division, showing PERFORMs forward and selected codes from either of the other two windows underneath each PERFORM. Also, Backward Tracing is available by clicking in the yellow highlighted checkbox and allows analysis of how any performed routine is referenced all the way back to PROGRAM-ENTRY. When Forward Tracing is currently starting at PROGRAM-ENTRY and not some other starting paragraph or section, then the yellow checkbox for doing backward tracing will not be shown.

All three windows together may be expanded and resized by expanding the CAT overall main window. Also the size of the Quick View Window may be adjusted vertically taking space from the Source Listing window above it. However, the Narrative window may not be resized beyond its 43 character maximum size. Also, Tabs are available to display other related data in the Quick View Window, to show other information such as Perform warnings, listings of syntax errors, and a selection of a various options.

Various drop-down boxes and buttons are available in appropriate points to make user control of the entire process very efficient and useful.

Alternate Compile Listing

The basic 'Alternate Compile Listing' produced by the DCD III on the mainframe is also available with one click and provides the same reports available on the mainframe.

```

Alternate Compile Listing
Goto Seq # Goto Following Search
USER PARM FIELD 06/07/2012 10:01 PAGE 01
DNF,FTB,LNC=047,

  PARM OPTIONS IN EFFECT FOR THIS PROGRAM
  ASKIPS      (FOR SKIP1, 2, 3) *****
  BF05       (MATCHES TO BUFNO) *
  NOBRT      -DO NOT BUILD BUS RULES * DCD III Release 3.8-1060 *
  CALL       -CALL REPORT *
  DYCALLS    (DYNAMIC CALL INFO) * -- A product of -- *
  CICS       (ALLOWED) *
  COPY      -COPY REPORT * MARBLE Computer,Inc. *
  DB2        (ALLOWED) * P.O. Box 920692 *
  DDC        -DATA DIV SCREF * El Paso, Texas 79902 *
  DLI        (ALLOWED) *
  ERRORS     Errors are printed *
  DNF        give Warnings for missing names * Phone: 1-800-252-1400 *
  FGCONSTS   -FIGURATIVE CONSTS * 1-915-845-0963 *
  FTB        * SEE NOTE BELOW * Fax: 1-915-845-7918 *
  NOIDD      NO DATA DIV LITERALS *
  INDEPENDENT (MODE) * Marble programs may not be *
  IREFS      INDIRECT REFERENCES * copied - Copyright 2012 *
  LITERALS   -LITERAL REPORT *
  LNCNT=047  (LINES PER PAGE) * Visit our Web Site ! *
  NONIS      No prog seq for PD Narrative * www.marblecomputer.com *
  NONUC      NARRATIVE-LOWER CASE *
  NONRSPACE  NO BLANK LINES *****
  PDC        -PROC DIV SCREF
  NOQUOTE    SINGLE APOSTROPHE
  RESOLVE    (COPY MEMBERS)
  SPRESOLVE  *OVERRIDE SUPPRESS
  SORTREG=00150000 (SORT REGION)
  SOURCE     -SOURCE LISTING
  SPREGS     -SPECIAL REGISTERS
  UNREF      (PRINT UNREF NAME)
  UO1        -UNUSED O1 RECORDS
  UDN        -UNUSED DATA NAMES
  UPARAS     -UNUSED PARA / SECTIONS
  UPTIONORE  (PARAS in Perform ranges)

1 ERROR MESSAGES 06/07/2012 10:01 PAGE 02
  
```

Alternate Compile Listing First Page

```

188 014400 LINKAGE SECTION.
189 014500 01 PARM-RECORD.
190 014600 05 PARM-COUNT PIC S9(4) COMP.
191 014700 05 FILLER PIC X(17).
192 014800 05 PARM-OPTION-VALUE PIC X(10).
193 014900
1SQ-NBR 1----6 PROGRAM-ID DEMOPGM 06/07/2012 10:01 R=1060 PAGE 13
195 #REG
1SQ-NBR 1----6 PROGRAM-ID DEMOPGM 06/07/2012 10:01 R=1060 PAGE 14
196 015100 PROCEDURE DIVISION USING PARM-RECORD.
197 015300 010-DO-HOUSEKEEPING.
198 015500 PERFORM P-TEST-PARM-FIELD.
199 PERFORM U1-STRING-FILE-NAMES.
200 MOVE ZEROS TO INV-M-ORIG-DT.
201 017100 020-OPEN-FILES.
202 017200 OPEN INPUT INV-MAST-FILE
203 017300 OUTPUT INV-MAST-FILE-2.
204 017900 030-READ-FIRST-RECORD.
205 018000 READ INV-MAST-FILE
206 018100 AT END
207 018200 MOVE 'Y' TO WS-END-OF-INV-FILE-SW.
208 018400 040-DO-ALL-RECORDS.
209 018500 PERFORM A-PROCESS-ALL-RECORDS
210 018600 UNTIL WS-END-OF-INV-FILE.
211 018800 050-CLOSE-FILES.
212 018900 CLOSE INV-MAST-FILE
213 019000 INV-MAST-FILE-2.
214 019200 999-COBACK. GOBACK.
1SQ-NBR 1----6 PROGRAM-ID DEMOPGM 06/07/2012 10:01 R=1060 PAGE 15
216 #REND
217 #REC

```

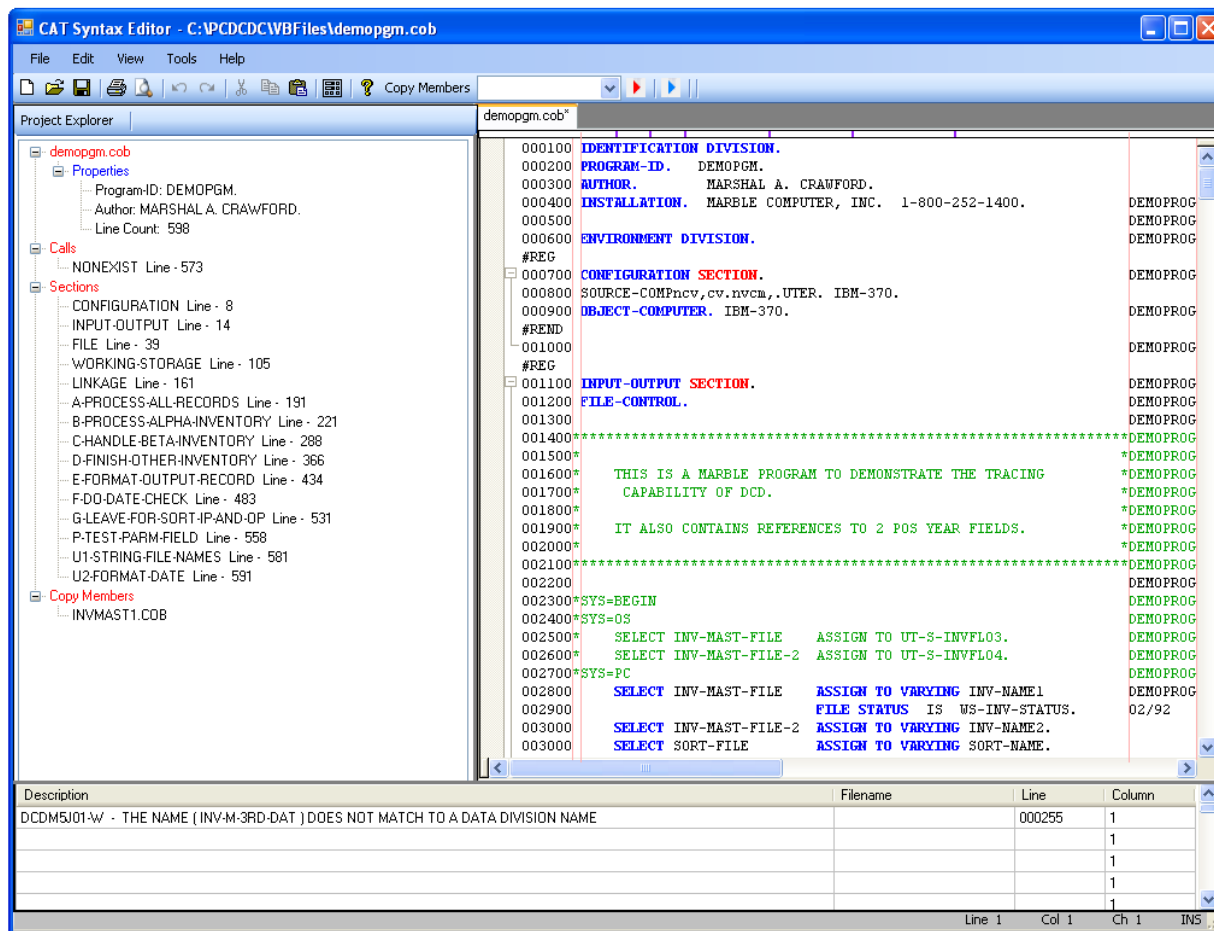
1-29 > Call 'NONEEXIST' using # (600)
Program-entry DEMOPGM using # (196)
Indirectly Tested @190,192
03/96 > If # < 18 (588)
Indirectly Used @189
3-19
DEMOPROG> If # = 'INVENTORY ' (594)
Indirectly Used @189
73----80 # REFERS TO DATA-NAME AT THIS SQ-NBR
73----80 SQ-NBR'S LOGIC FLOW
DEMOPROG 189 BEGIN OF PROGRAM
DEMOPROG
DEMOPROG PERFORM 585 THRU 605
PERFORM 608 THRU 616
120
40
72
DEMOPROG
44
143
DEMOPROG PERFORM 218 THRU 244
144
40
72
DEMOPROG GOBACK
73----80 SQ-NBR'S LOGIC FLOW

Sample Listing with some Data and Procedure Division

While CAT offers a much enhanced format in the main CAT window and with increased flexibility from the ACL (Alternate Compile Listing), there may be times where the ACL comes in very handy for special viewing of its many reports and also for its unique source listing with added Narrative.

Cobol Syntax Editor

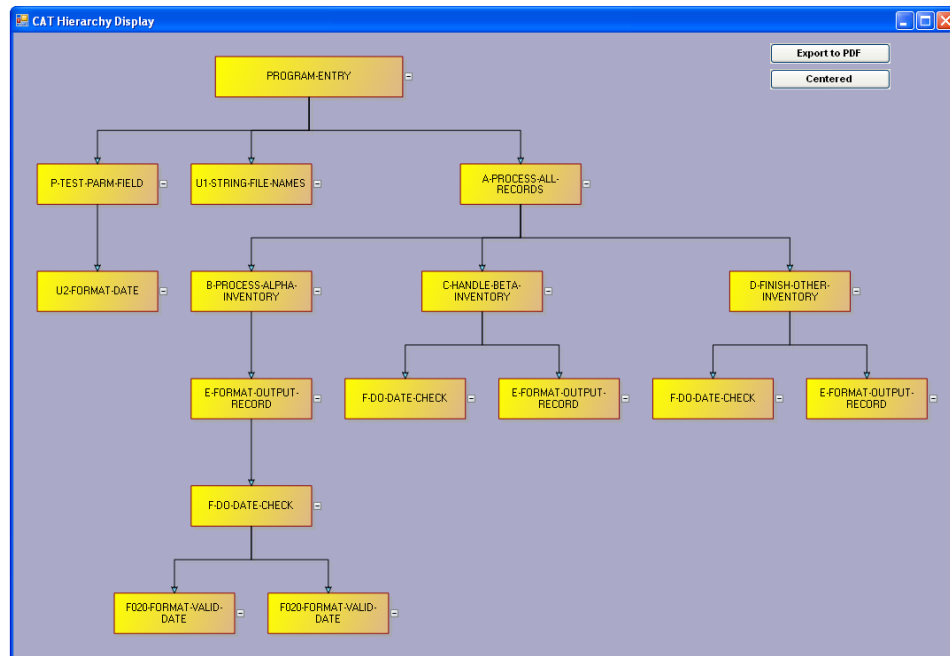
Another major feature of CAT is its Syntax Editor. CAT's source code editor, working together with CAT's analysis tools, provides the ability to modify programs as they are understood by a technician – or to write them from scratch. The editor has its own window – separate from the main CAT window and the ACL window.



Syntax Editor

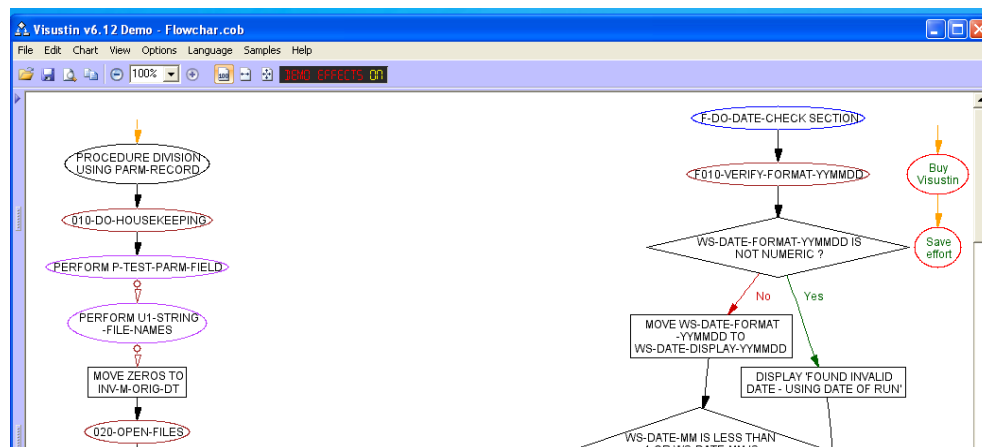
Charting

CAT provides both hierarchy charts and flowcharts to give a bird's eye perspective on programs. Hierarchy charts give a straight-forward view of program organization based on the analysis of COBOL PERFORMs. Hierarchy charts may begin at any PERFORM or at the start of the program.



Hierarchy Chart

Flowcharts show the internal workings of a program in greater detail.

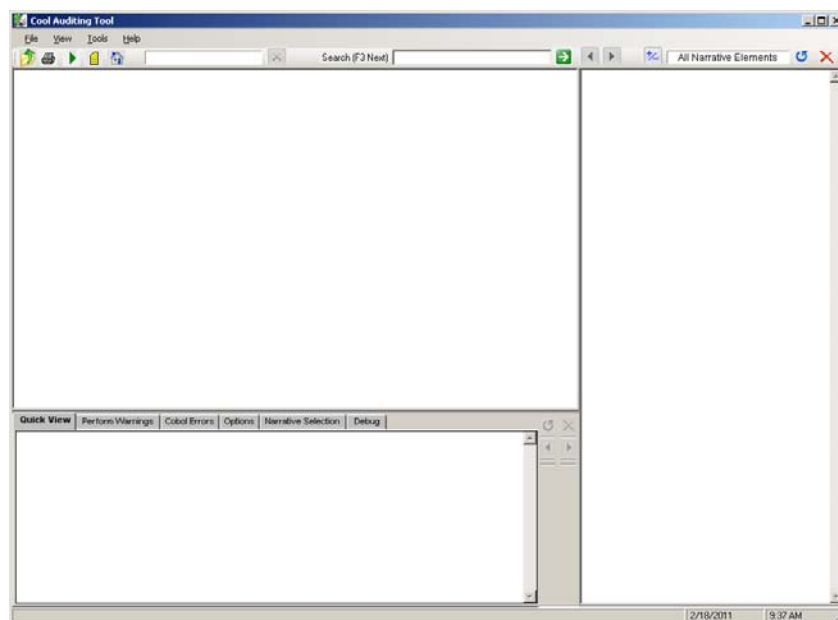


Flowchart

Overview

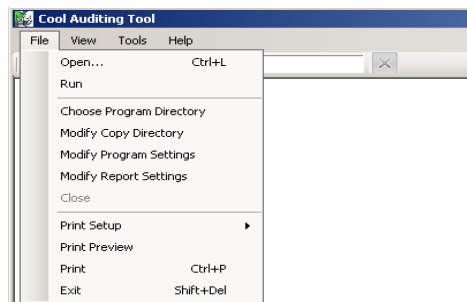
Overview of Cool Auditing Tool for COBOL

This overview provides a quick understanding of the power of 'Cool Auditing Tool' (CAT) for Cobol and allows the user to begin to see how some of the features available within CAT may significantly assist her or him in tasks related to the understanding and maintenance of COBOL programs. We use a Demo Program provided with the installation of CAT to assist in highlighting main features. The user is recommended to run this Demo Program while following through this overview. Below is a basic view of CAT when the program loads.



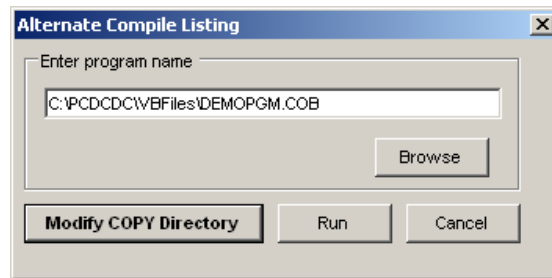
CAT Main Display

To process a COBOL program, click on File to display a dropdown menu as shown below:



File drop down menu

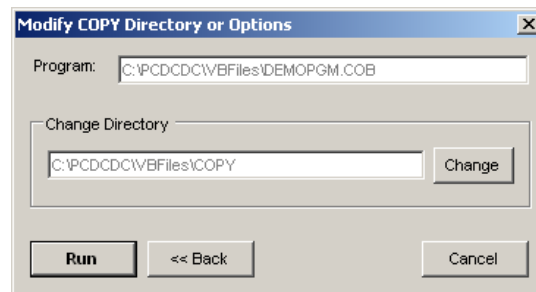
Now click on the Open button to browse for a COBOL program, as shown in the next image:



Dialog to Select COBOL program

The Test program we will use resides on the directory to which CAT was installed. If the directory shown above is not the install directory, determine what the install directory was and go to the VBFiles of that folder and select DEMOPGM.COB there.

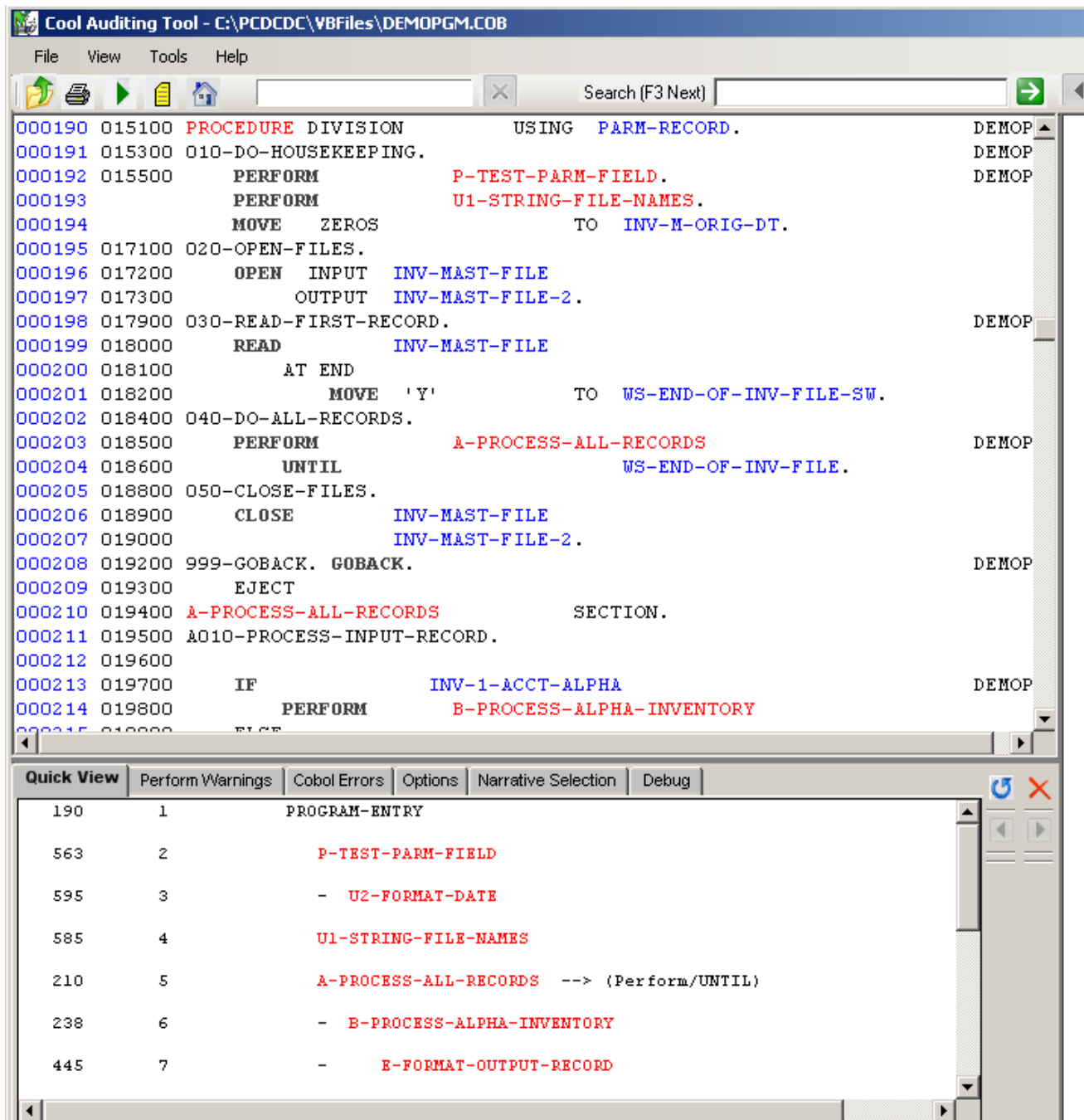
When the Demopgm is correctly selected, click on the 'Modify COPY Directory' button to bring up the next window.



Dialog to Select COBOL Copy Members

If the install directory was installed to the folder shown, then the COPY directory will be correct for DEMOPGM. If not, use the Change button and go to the same install directory used above and click on the VBFiles folder within it and then click on the COPY folder. When ready, click Run to do analysis on this program and to fill in two of the three main CAT windows. The third window is only filled in by user interaction after the first two windows are filled in.

After a second or two of processing, the left side of the CAT application will look as shown on the next page:



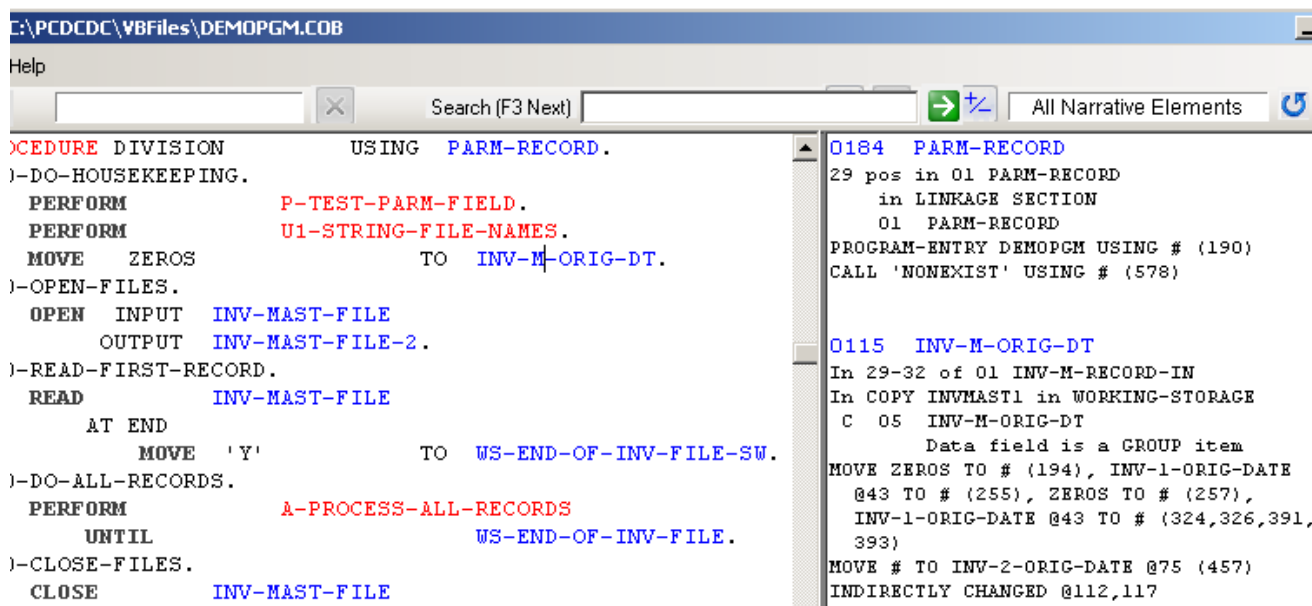
Display on left side after doing Run

The top window represents the COBOL program. On the very left side of this window are the expanded sequence numbers (after COPYs have been resolved) to identify each COBOL line. Notice that the scroll bar has been adjusted so that the COBOL program starts at the beginning of the Procedure Division. Also, notice that data names are highlighted in **Blue** and PERFORMed Paragraphs are highlighted in **Red**.

Looking at the bottom window, the tab for the **Quick View Window** is highlighted. This window gives a picture of forward and correctly indented PERFORMs beginning from any point in the program and currently is set to begin from the start of the Procedure Division. Also, notice that the Perform of A-PROCESS-ALL-RECORDS is marked as a Perform/Until for help in following and understanding program logic.

The window just mentioned, the Quick View Window, is also used as a backward tracing window allowing full forward / backward tracing while viewing, auditing and maintaining a COBOL program and this feature is explored further in the next page or two and then later in this manual.

Now let's focus on the top window again, do a single click on the **blue** highlighted name **PARM-RECORD** on the first line and then move the mouse over the name **INV-M-ORIG-DT** highlighted in **blue** 5 lines down from the top. Do a single click on this name also. The window on the right is now in use to show information on these data names as shown in the image below: *Note – if Narrative options have been changed, what is shown below may differ slightly in format.*



Display Narrative on right side after Selecting a field

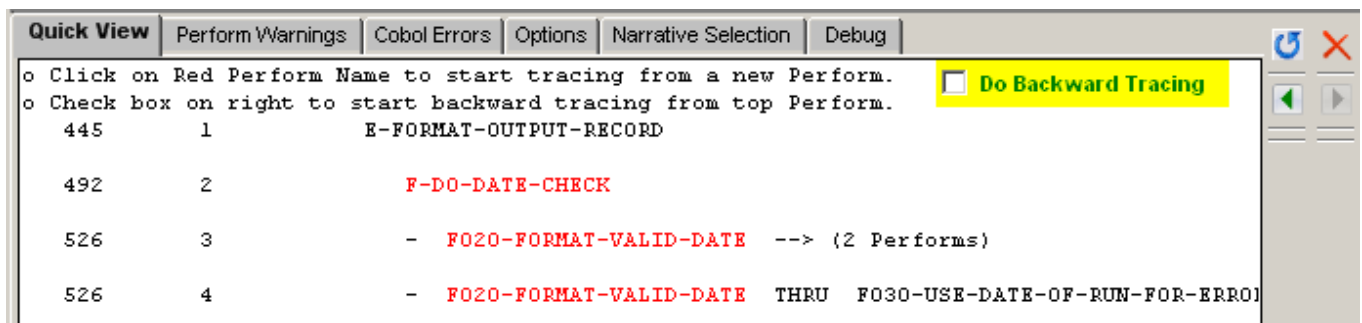
The narrative information provided on the first line identifies the name **INV-M-ORIG-DT** along with expanded sequence number of where the name resides in the program are in **blue**. Also provided on the second line is the 01 record that the field belongs to and with (FROM – TO) positions where the field resides in the 01 record.

On the third line, this field is shown as being in COPY INVMAS1 and the record (and associated field) is in Working Storage. The next two lines mimic how the field might appear in the Data Division and indicate that this field is a GROUP field.

After that, PROCEDURE DIVISION references (discussed more under a different topic) show all direct references to this field in a format designed to resemble PROCEDURE DIVISION verbs that are used.

The last line shows Indirect References. For instance, there is a MOVE SPACES TO the 01 record INV-M-RECORD-IN which **Indirectly Changes** this field indirectly at line **112**. A different Indirect reference is done at line **117**.

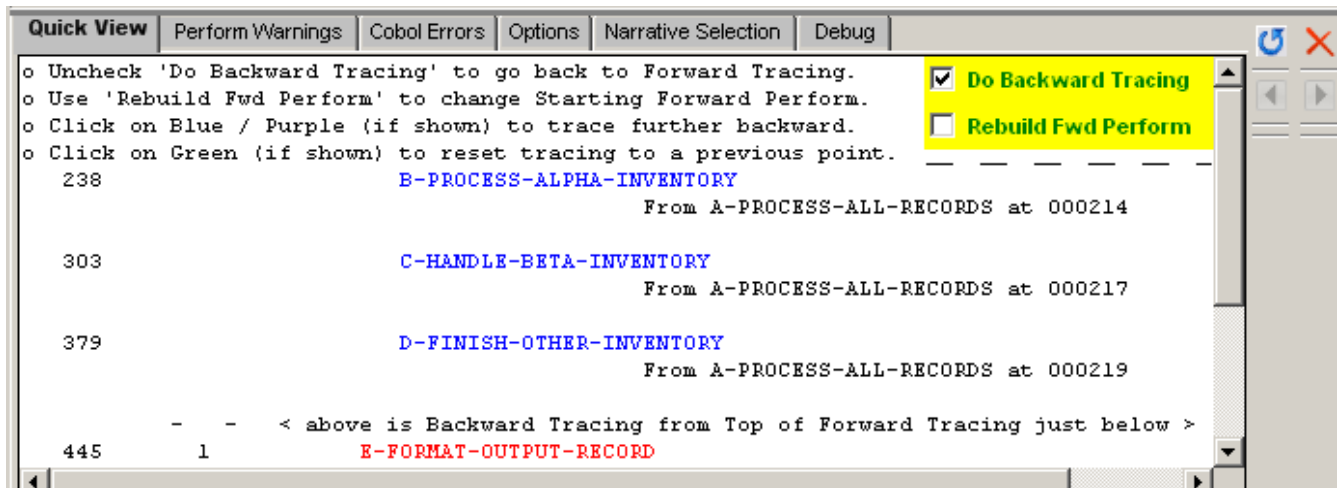
More on COBOL Narrative in discussed under a different topic. Now let's go back to the Quick View Window on the bottom of the left side and move the mouse over the indented section name E-PROCESS-OUTPUT-RECORD, and do a single click on the name highlighted in **red** to modify the Quick View Window on the bottom of CAT. The Quick View Window after single clicking appears as shown below:



Quick View Window doing Forward Tracing

Notice three immediate differences from the earlier **Quick View Window** as follows:

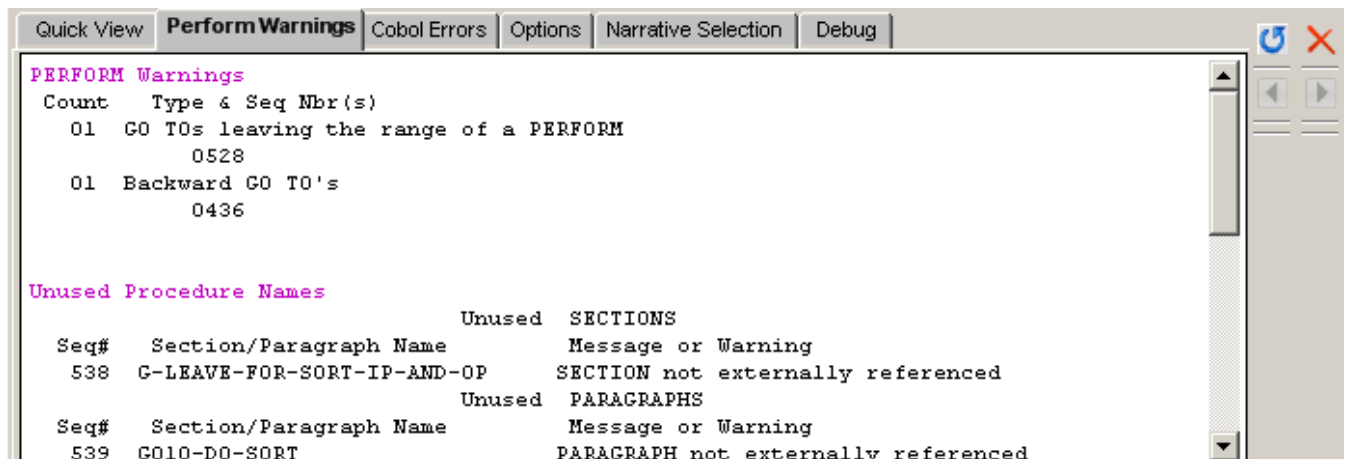
- The Quick View Window now starts at the routine E-FORMAT-OUTPUT-RECORD
- There are two comments lines giving actions that may be done here within forward tracing
- There is a yellow highlighted check box on the right marked 'Do Backward Tracing'. Checking this check box will bring up the following window shown on the next page to start backward tracing from this performed section E-FORMAT-OUTPUT-RECORD



Quick View Window doing Backward Tracing

A more detailed description of tracing with forward and backward PERFORMs is done in the section 'Quick View Window'.

Now let's cover some of the other tabs in this bottom window. Click on the **Perform Warnings** tab and notice the following image appears:

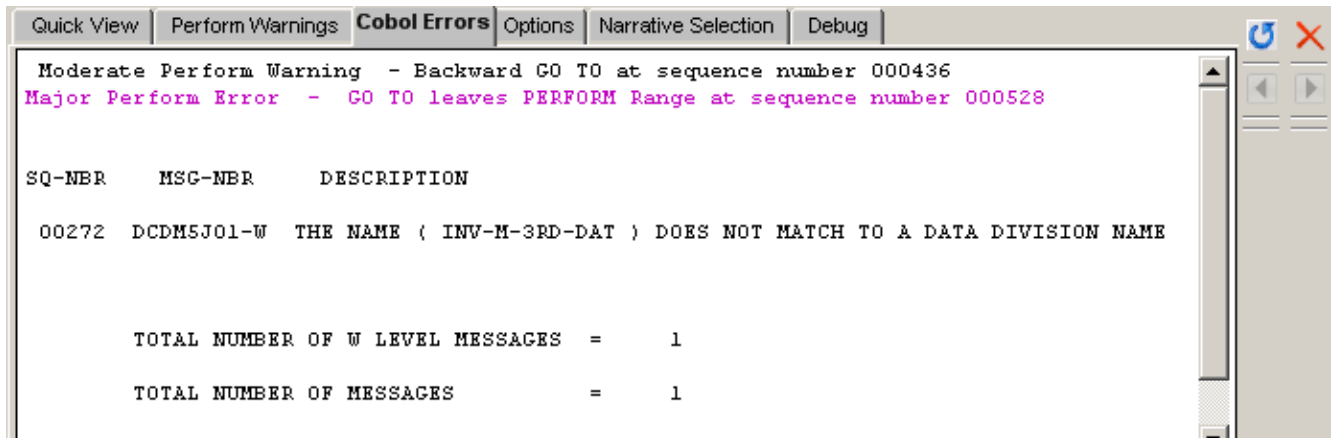


Perform Warnings

This window shows four types of information covered fully under another topic. For now, what is visible on the window shows **PERFORM Warnings** for two PERFORMs, one containing a GO TO leaving the range of the PERFORM and another listing a Backward GO TO.

What else is visible on this window is the beginning of an **Unused Procedure Names** report. This report shows unreferenced code or potentially dead code. The user, typically after viewing this report, may want to examine the program carefully and see why there is dead code in the program.

Now let's click on the **Cobol Errors** tab to produce the Cobol Errors window:

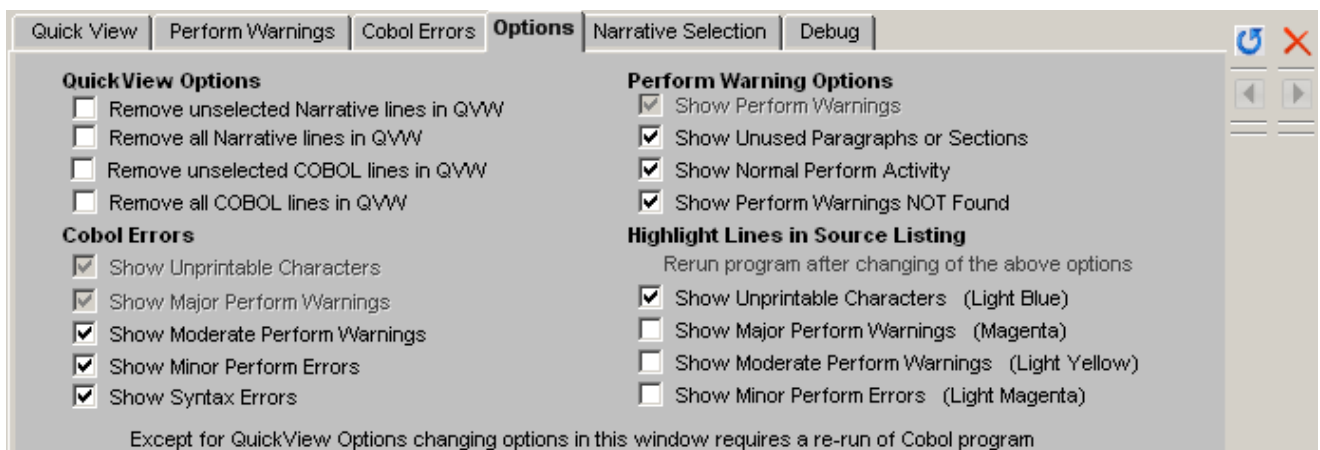


COBOL Errors

Any Perform Warnings shown in the previous report are categorized as Major, Moderate, or Minor and are listed before other errors in this window.

Below that are Syntax Errors, such as data names not found. In this case, we left one data-name unmatched and a Warning Message appears for this name INV-M-3RD-DAT.

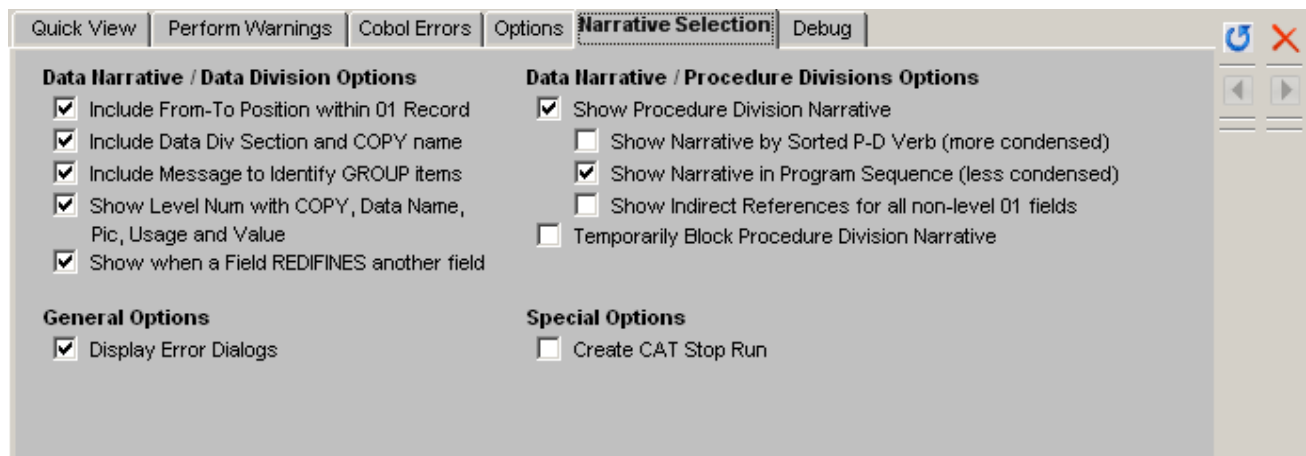
Go past the tab for **Run Errors** and now Click on the tab **Options**. The following window appears:



Options Tab

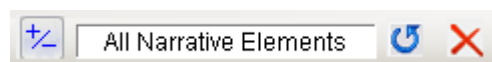
These options allowing modifying the behavior of how CAT does or does not fill in various items on the left side of CAT both top and bottom windows. These options are discussed more fully in another topic.

The next tab we will look at is **Narrative Selection**. Click on this tab and the following window appears:



Narrative Selection Tab

These options along with the following [+/-] button from just above the Narrative window on the right hand side of CAT control how Data Division Narrative is formatted for the user to see.



Narrative Formatting control

Again, these options are discussed more fully in another topic.

There is a Code Compare utility for comparing source code included and that will be covered under another topic.

Another feature we will discuss in this CAT Overview is the 'Alternate Compile Listing'. The Alternate Compile Listing is a feature, along with its parent software package DCD III, offered by Marble Computer and has been around on the mainframe for over 35 years, and also it has been heavily enhanced during that time. The Alternate Compile Listing available on the mainframe is available here in a much nicer format, as it can be viewed entirely (all 132

characters) rather than scrolling to the left and right using a mainframe or 3270 screen. It can be very useful in many instances where a programmer decides to view that listing with one mouse click, rather than following CAT Windows and allow the user to come back to the CAT windows when specific information is required. Find the following button on the tool bar and click on it.



Alternate Compile Listing button

The Listing below is what initially comes up as a separate window:

```

Alternate Compile Listing
-----
USER  PARM  FIELD      06/07/2012 10:01  PAGE  01
DNF,FTB,LNC=047,

  PARM OPTIONS IN EFFECT FOR THIS PROGRAM
  ASKIPS      (FOR SKIPL, 2, 3)
  BFO5        (MATCHES TO BUFNO)
  NOBRT       -DO NOT BUILD BUS RULES
  CALL        -CALL REPORT
  DYCALLS     (DYNAMIC CALL INFO)
  CICS        (ALLOWED)
  COPY        -COPY REPORT
  DB2         (ALLOWED)
  DDC         -DATA DIV SKREF
  DLI         (ALLOWED)
  ERRORS      Errors are printed
  DNF         give Warnings for missing names
  FGCONSTS    -FIGURATIVE CONSTS
  FTB         * SEE NOTE BELOW
  NOIDD       NO DATA DIV LITERALS
  INDEPENDENT (MODE)
  IREFS       INDIRECT REFERENCES
  LITERALS    -LITERAL REPORT
  LNCNT=047   (LINES PER PAGE)
  NONIS       No prog seq for PD Narrative
  NONUC       NARRATIVE-LOWER CASE
  NONRSPACE   NO BLANK LINES
  PDC         -PROC DIV SKREF
  NOQUOTE     SINGLE APOSTROPHE
  RESOLVE     (COPY MEMBERS)
  SRESOLVE    *OVERRIDE SUPPRESS
  SORTREG=00150000 (SORT REGION)
  SOURCE      -SOURCE LISTING
  SPRNGS      -SPECIAL REGISTERS
  UNREF       (PRINT UNDEF NAME)
  UO1         -UNUSED O1 RECORDS
  UDN         -UNUSED DATA NAMES
  UPARAS      -UNUSED PARA / SECTIONS
  UPICIGNORE   (PARAS in Perform ranges)

*****
* DCD III  Release 3.8-1060
* -- A product of --
* MARBLE Computer,Inc.
* P.O. Box 920692
* El Paso, Texas 79902
* Phone: 1-800-252-1400
* 1-915-845-0963
* Fax: 1-915-845-7918
* Marble programs may not be
* copied - Copyright 2012
* Visit our Web Site !
* www.marblecomputer.com
*****

1  ERROR MESSAGES      06/07/2012 10:01  PAGE  02
  
```

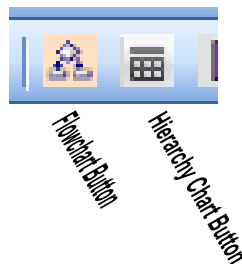
Alternate Compile Listing Options Page

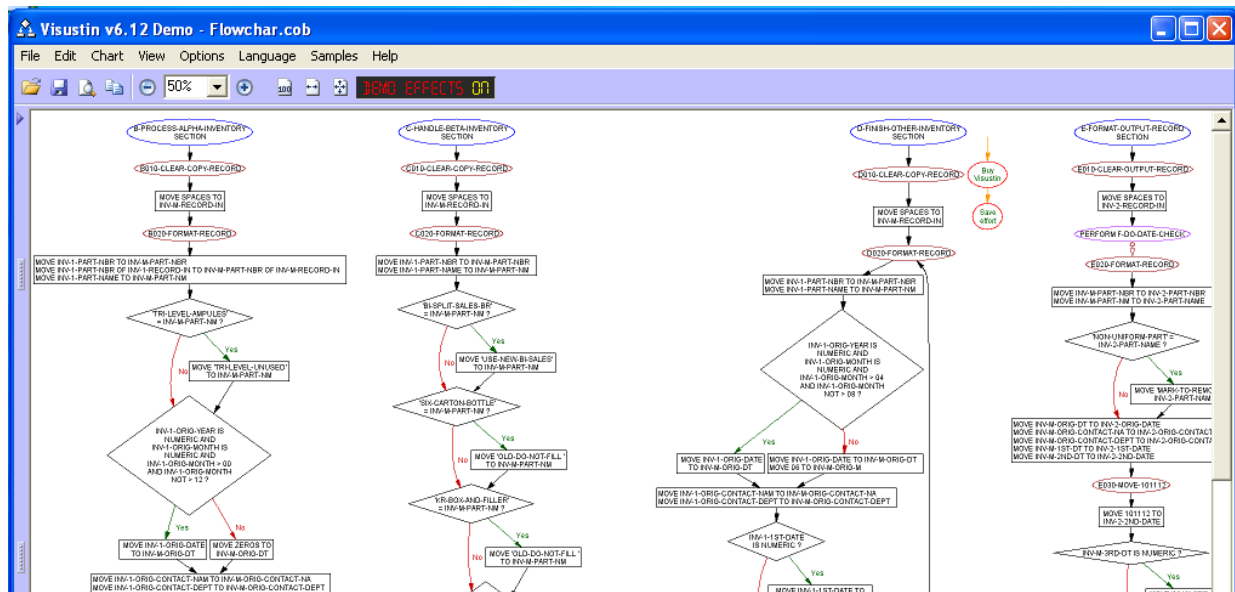
Use the scroll bar to get beyond the options page and to slowly visit a few of the different pages, in both the Data Division and the Procedure Division, and to get a feeling for the Alternate Compile Listing. The following window below is in between the DATA and PROCEDURE divisions and shows features of both.

Alternate Compile Listing Source Listing

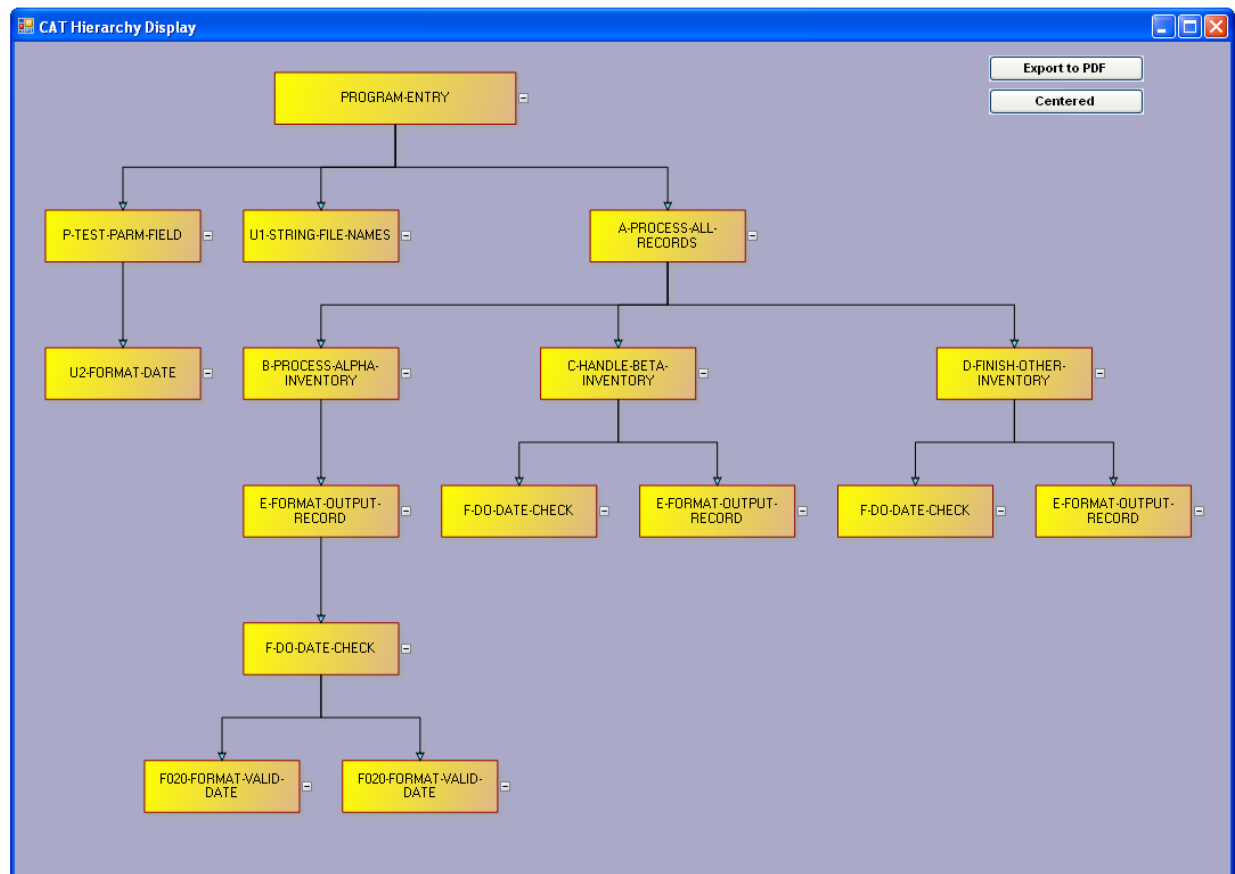
Also, go to the end of the Alternate Listing and view condensed reports which include, COPY members included, CALLs made, Literals, a Condensed Data Division report, a Condensed Procedure Division report, Unused 01 Records report, Unused Data Name Report, and Unused Paragraphs / Sections.

Rounding off CAT's capacity as a COBOL analysis package are two types of charts – hierarchy charts and flowcharts. CAT's hierarchy charts serve to show a program's structure based on its PERFORMs. Flowcharts show the inner workings of a program in greater detail than the hierarchy charts, describing structures such as decision making, specific commands and transfer of control. Note that use of flowcharts requires the purchase of additional software. Flowcharts and hierarchy charts are accessible via two buttons in the CAT graphical interface.





Flowchart



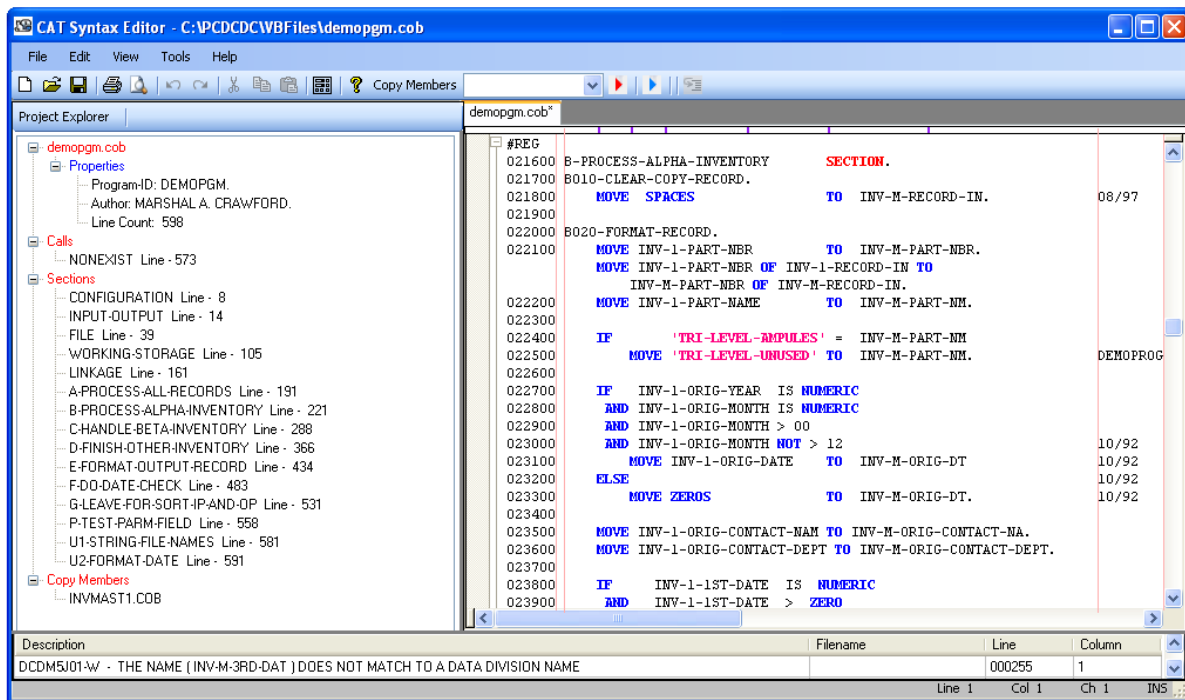
Hierarchy Chart in Top-to-Bottom Format

CAT also includes a source code editor, which complements CAT's analysis tools – and contains some of its own – making CAT a robust package for auditing, maintaining and developing COBOL. To launch the editor, click on this button in the CAT main display.



Syntax Editor Launch Button

The Syntax Editor operates in its own window, distinct from the main CAT main display, the ACL and the chart windows. This window contains three main panes – one to the left that lists important elements of the program, one to the right that contains the actual source code, and one at the bottom that lists errors found in the source code.



Source Code Editor

This ends the Overview for Cool Auditing Tool for COBOL.

Quick View Window (QVW)

Located at the bottom of the CAT display is a multi-function Tab Display. This display has several user selectable windows including the Quick View, Perform Warnings, Cobol Errors, Run Errors and other displays that are more fully described in the Tab Options section of this user guide.

Tabs just above the QVW



Display of Tab Names

QVW Forward Tracing

Complete tracing is available, whether **forward** or **backward** within the Quick View Window.

The algorithms used for forward tracing and backward tracing are similar in the following way:

- They both start at the currently selected PERFORM name
- They both use one level of indenting to show the immediate relationship of PERFORMING from one routine to the next either forward or backward

How the algorithms differ is shown below first for the Forward Tracing and then for Backward Tracing along with interaction between forward and backward tracing. At the end of the examples provided is a complete summary description of how these two modes (Forward and Backward) tracing differ.

Following a Run command, the 'Quick View' tab is defaulted to. This window is designed to work with the other two main windows in following the logic of a COBOL program. Each time a program is brought into Cool Auditing Tool, the Quick View Window starts with forward tracing from the top of the Procedure Division. Forward Tracing is covered in detail in the next couple pages of this manual and then Backward Tracing is covered in detail after that.

First, let's discuss forward PERFORMs from any starting point. In the following illustration, the Quick View window is started at the beginning of the Procedure Division when the program is first run and the source window above it is also positioned at the beginning of the Procedure Division.

Quick View	Perform Warnings	Cobol Errors	Options	Narrative Selection	Debug
190	1	PROGRAM-ENTRY			
563	2	P-TEST-PARM-FIELD			
595	3	- U2-FORMAT-DATE			
585	4	U1-STRING-FILE-NAMES			
210	5	A-PROCESS-ALL-RECORDS --> (Perform/UNTIL)			
238	6	- B-PROCESS-ALPHA-INVENTORY			
445	7	- E-FORMAT-OUTPUT-RECORD			
492	8	- F-DO-DATE-CHECK			
526	9	- F020-FORMAT-VALID-DATE --> (2 Performs)			
526	10	- F020-FORMAT-VALID-DATE THRU F030-USE-DATE-OF-RUN-FOR-ERROR			
303	11	- C-HANDLE-BETA-INVENTORY			
(see # 8)		- F-DO-DATE-CHECK --> (3 Performs)			
(see # 7)		- E-FORMAT-OUTPUT-RECORD --> (3 Performs)			
379	12	- D-FINISH-OTHER-INVENTORY			
(see # 8)		- F-DO-DATE-CHECK --> (3 Performs)			
(see # 7)		- E-FORMAT-OUTPUT-RECORD			

Starting at Procedure Division begin

Following the above example, within the first mainline routine in the PROCEDURE DIVISION, there are PERFORMs to only 4 routines as shown below:

P-TEST-PARM-FIELD

U1-STRING-FILE-NAMES

U2-FORMAT-DATE

A-PROCESS-ALL-RECORDS

See the four indented PERFORMs under the PROGRAM-ENTRY. Notice the comment after A-PROCESS-ALL-RECORDS indicating that this PERFORM is a **Perform/UNTIL**.

Notice further that there are exactly three PERFORMs indented one level under A-PROCESS-ALL-RECORDS, namely B-PROCESS-ALPHA-INVENTORY, C-HANDLE-BETA-INVENTORY and D-FINISH-OTHER-INVENTORY.

Let's stop here briefly and note the two numeric fields on the left side of the PERFORM names.

The first one is the expanded sequence number of where the PERFORM resides in the source listing above it. We use the term ‘expanded sequence number’ as opposed to just sequence number because the source listing usually has COPY members in it that have been added and the sequence numbers are expanded to include the COPY records within the COPY members that are brought into the listing.

The second number is a simple in sequence number showing the PERFORM number. This number is needed to keep the Quick View window simple. Look at PERFORM number 8 which is a PERFORM of F-DO-DATE-CHECK and notice that it has two Perform Paragraphs beneath it (one with 2 PERFORMs documented as being inside of F-DO-DATE-CHECK) and one which is a PERFORM THRU. Now notice the next to last line in the image which has a comment on the left side as follows:

(see # 8)

and then shows a PERFORM of F-DO-DATE-CHECK and documents 3 such PERFORMs underneath a PERFORM of D-FINISH-OTHER-INVENTORY.

Rather than once again showing the two unique PERFORMs underneath line 8 at line 9 and 10, this (see # 8) limits the size of the Quick View window. This is also important when including other selected codes relevant to the user’s current interest. If several lines are included under the PERFORMs at lines 8, 9 and 10 they will be limited to being shown once, rather than listing them every time a PERFORM is performed from elsewhere.

Adding other COBOL instructions into the QVW

There are two ways to include extra code underneath the current range of PERFORMs in the Quick View window.

- Add lines of code from the Procedure Division of the source listing. This is done by clicking on one or more COBOL verbs in the source listing, and then clicking on the ‘Refresh’ button to the far right of the Quick View Tab. (COBOL Verbs are bolded in the Procedure Division for easy recognition, and they will change color after being clicked on). Once lines are added they may be removed by re-clicking on the verb until the color turns back to black, and then again click on the ‘Refresh’ button to remove them.

NOTE - This ‘first way of adding code’ feature is currently being finalized and not available in this release of the software.

- The second way of adding code to the Quick View Window is to bring in all Procedure Division code that is currently selected in the Narrative Window.

Code that is selected in the Narrative Window is built by clicking on Data-Names which are all highlighted in **blue** in the source listing. When a data field is clicked on, all Procedure Division code using that data-name or field have Narrative created to show all references to that field

If Narrative is built for one or more data-names, then just clicking on the 'Refresh' button to far right of Quick View Tab will create a closely identical form of the Narrative, and merge these lines by Procedure Division line number into the Quick View Window. These lines are indented to match the indentation of the Perform range that it belongs to.

When other lines of code are selected and they do not fall within any of the selected performed Paragraphs or Sections, then these lines of code will be directed to the bottom of the Quick View Window and marked there as ' - - - - End Perform Range - - - - following code not within selected PERFORMs - - - - '.

NOTE - See 'Options Tab' for further controlling of the selection of these two types of lines.

Restarting the QVW to start at a different PERFORM

Clicking on a Paragraph or Section Name where ever it is highlighted in **red**, will force a complete refresh of the Quick View Window and the window will now begin with the Paragraph or Section that has been clicked on.

Forward / Backward buttons are also available on right side of Quick View Window as arrows and when they are highlighted in **green** they may be used to go backwards to the previously viewed beginning paragraph or section, OR to return to paragraph or section viewed just before going backwards.

Also, a Home button is available at top of CAT window for bringing the CAT window back to the start of the Procedure Division.

If a Paragraph Name or Section Name is not highlighted in **red**, it indicates there is no PERFORM for that Paragraph or Section and the paragraph/section name may not be clicked on.

Within the use of a COBOL SORT verb, the use of INPUT PROCEDURE and OUTPUT PROCEDURE are treated as COBOL PERFORMs and when present are treated as PERFORMs.

Enhanced detail within the QVW Display

Counting of PERFORMs is done to identify how many times the PERFORM exists within the range of the higher up PERFORM that it belongs to.

If there is only one PERFORM within the range being presented, a count is not identified. Otherwise an exact count of all PERFORMs is given.

PERFORMs are broken into four categories as follows:

PERFORM (with or without THRU option)
PERFORM VARYING
PERFORM UNTIL (with NO VARYING clause)
PERFORM nnn TIMES

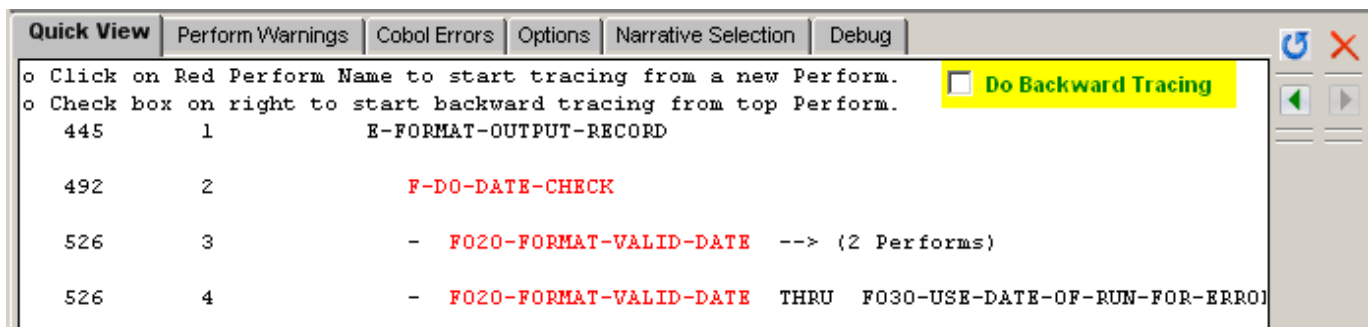
The type of PERFORM is identified to the right of the PERFORM name, except for the first typical type of PERFORM without the VARYING, UNTIL, or TIMES clause. When there are multiple types of PERFORMs in any one range, besides an overall count of the number of PERFORMs, there will be a count for each category of VARYING, UNTIL, or TIMES following the main count.

Indenting is done back and forth. Excessive indenting will be stopped from going further if there is not room to keep indenting.

QVW Backward Tracing

Backward Tracing is available from any PERFORMED routine within the program, with the exception of PROGRAM-ENTRY as it is impossible to go backward from the beginning of the program.

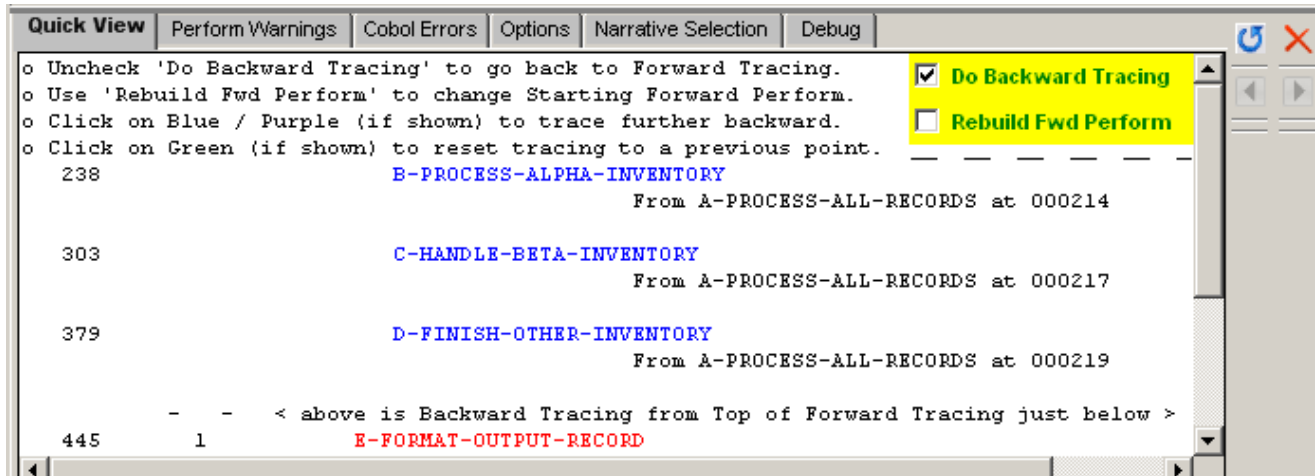
The following image shows forward tracing from a routine labeled E-FORMAT-OUTPUT-RECORD. Notice the checkbox available highlighted in yellow named 'Do Backward Tracing'.



Using yellow highlighted checkbox to invoke Backward Tracing

The two comments on the first two lines basically are there as a form of help. Clicking on the checkbox to do backward tracing is used to do backward tracing from the routine E-FORMAT-OUTPUT-RECORD. Using this checkbox is designed to answer the

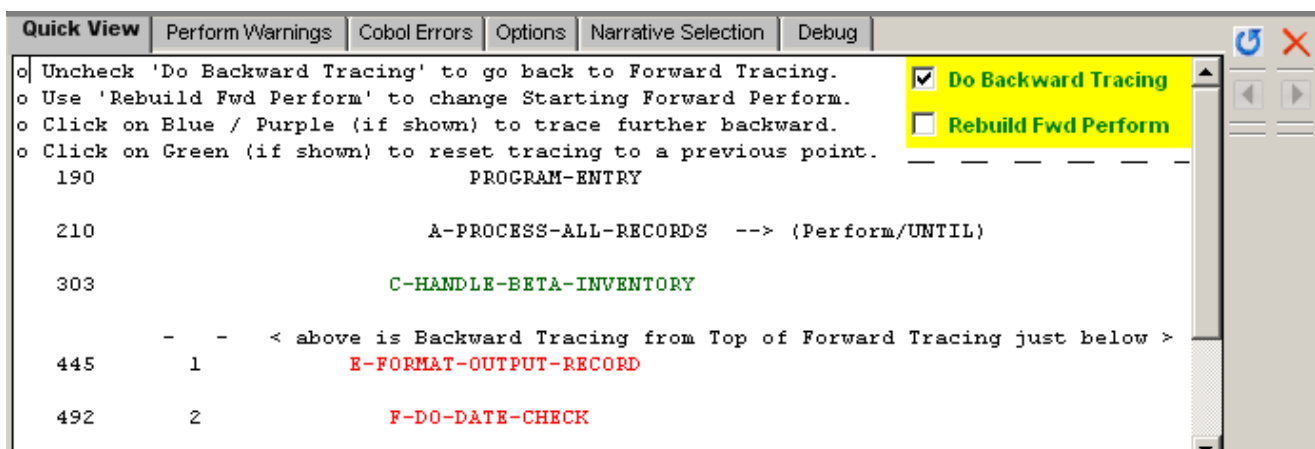
question, 'How does program logic get to this routine.' After checking this box a new window appears:



Tracing back to Program-Begin OR to more than one routine performing a PERFORM

The four comments on the first four lines are also a form of help to assist in either doing backward tracing or showing a couple of different ways of continuing forward tracing. Notice that in the image above, there are 3 PERFORMED routines that PERFORM E-FORMAT-OUTPUT-RECORD and they are all highlighted in BLUE. Underneath each blue highlighted routine is a comment line showing one or more lines showing what routines are used to perform that routine. A maximum of 3 comment lines is allowed underneath each BLUE highlighted routine. If there are more than 3 lines, a message will indicate the presence of excess routines.

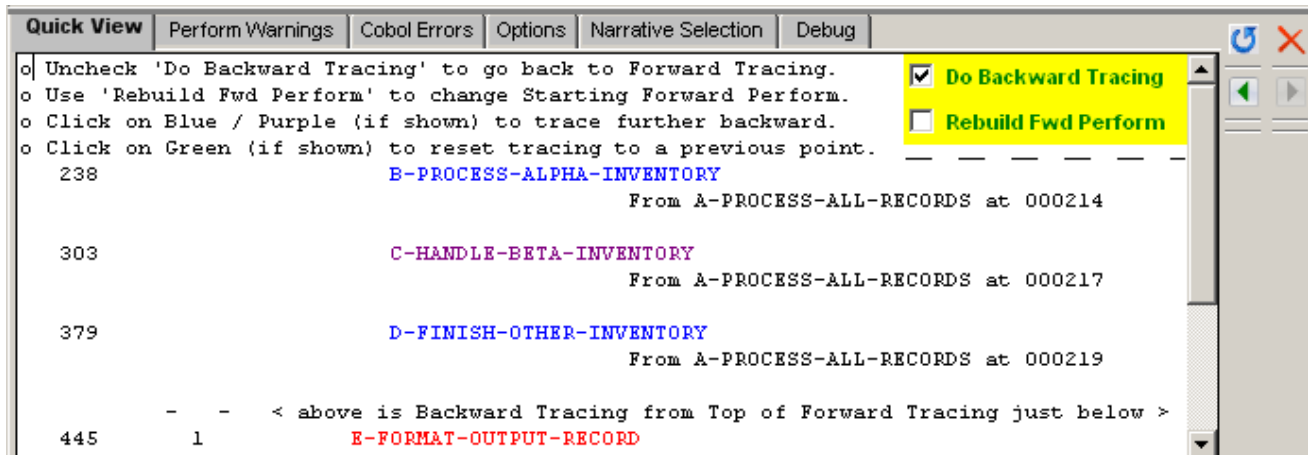
Clicking on any one of the BLUE highlighted routines will follow the logic back from there. For instance, clicking on the middle BLUE routine C-HANDLE-BETA-INVENTORY will continue backward tracing to the following window:



Tracing further back from a previously **blue** highlighted PERFORM

Notice that in tracing backwards, the routines B-PROCESS-ALPHA-INVENTORY and D-FINISH-OTHER-INVENTORY have been removed as they are not now being traced and the routines all the way back to PROGRAM-ENTRY are now shown. C-HANDLE-BETA-INVENTORY is now highlighted in GREEN and we will cover that significance in a moment. If there were other multiple PERFORMs leading to any one PERFORM, then they would be shown and the user would need to click on those to continue to reach PROGRAM-ENTRY. In this case there were no more multiple PERFORMs in the path back to the beginning of the PROCEDURE DIVISION.

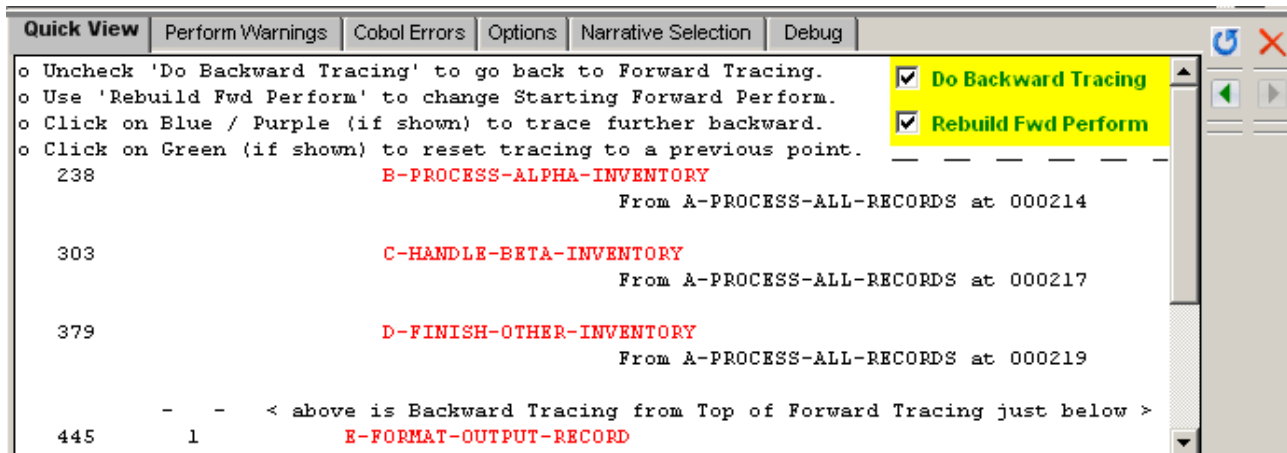
Now that the path to PROGRAM-ENTRY is found for C-HANDLE-BETA-INVENTORY, the technician would normally be interested in following the other formerly highlighted BLUE routine to trace them backwards and to do that, we need to go back to the previous window and that is accomplished by clicking on the GREEN highlighted perform C-HANDLE-BETA-INVENTORY which brings up the previous window slightly modified as seen below:



After doing a **RESET** from Tracing further back

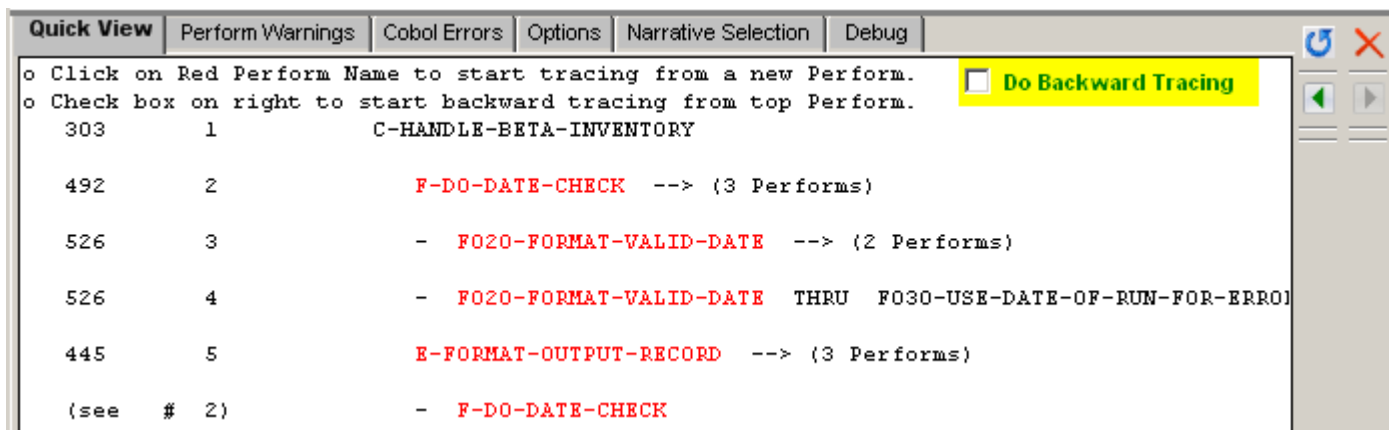
Notice the C-HANDLE-BETA-INVENTORY is highlighted in PURPLE to indicate that that routine has already traced. Clicking on B-PROCESS-ALPHA-INVENTORY or D-FINISH-OTHER-INVENTORY will act in a manner similar to what was shown for C-HANDLE-BETA-INVENTORY.

Lastly, notice the new highlight in yellow checkbox labeled 'Rebuild Fwd Perform'. The purpose of this checkbox is to allow the user a way of restarting forward tracing at one of the routines in backward tracing, by highlighting these routine names in **RED**, and then allowing a click on that routine to rebuild forward tracing from that point. Clicking on the 'Rebuild Fwd Perform' checkbox will bring up the following window:



Preparing to restart Forward Tracing from a point within Backward Tracing

Notice that all routines are highlighted in RED, allowing the user to start forward tracing from any of the routines shown. If the user then clicks on C-HANDLE-BETA-INVENTORY, then forward tracing will begin from that point as shown in the following window:



Forward Tracing restarted from a previous Backward Tracing PERFORM

From this point, the user may view forward tracing from C-HANDLE-BETA-INVENTORY, may click on another routine to start forward tracing from there, or may check the checkbox 'Do Backward Tracing' to start backward tracing at the currently positioned routine of C-HANDLE-BETA-INVENTORY. As shown in the above examples, the forward and backward tracing algorithms are different from each other. The following describes the unique differences between forward and backward tracing:

What is Unique to Forward Tracing

- When a new PERFORM name shown in **Red** is clicked on, Forward Tracing is branched to even if the Quick View Window currently was in Backward Tracing (was previously indicated by being checked for 'Do Backward Tracing')
- Forward Tracing is indented downhill one level to show what PERFORMs are performed within a given routine as far as needed to complete each trace through multiple PERFORMs
- When a Perform with possible other PERFORMs underneath it have been expanded once previously, a (see #) line is included to keep the report smaller
- Narrative and COBOL lines are available by option to be included in forward tracing underneath the respective PERFORMs

What is Unique to Backward Tracing

- Backward Tracing is indented uphill one level
- Ultimately, backward tracing continues to PROGRAM-ENTRY or PROCEDURE DIVISION begin, however, when a PERFORMED routine is called by more than one routine, tracing is stopped there with **BLUE** highlighting to show each of the multiple routines that perform the routine below it, and this allows the programmer/technician to click on one of these PERFORMs at a time to continue tracing that path further back, ultimately to PROCEDURE DIVISION begin
- After clicking on a **BLUE** perform, the other **BLUE** Performs disappear, the previously **BLUE** Perform becomes **GREEN**, allowing a RESET to go back to the previous window showing all the **BLUE** routines just shown with the routine just traced turning **PURPLE** in color to indicate that the PERFORM was traced
- Tracing on one **BLUE** perform either then goes to PROCEDURE DIVISION begin (noted as PROGRAM-ENTRY) or stops again at another place where there are duplicate PERFORMs reaching one PERFORM routine allowing the programmer / technician to continue tracing from there
- When a **GREEN** highlighted Perform is clicked on, besides going back to the multiple **BLUE** highlighted PERFORMs window, the **BLUE** highlighted PERFORM that was clicked on turns **PURPLE** in color to indicate that PERFORM has previously been traced. Also, the previous uphill tracing just shown disappears to allow clear tracing uphill or backwards on another selected PERFORM
- A checkbox for 'Rebuild Fwd Perform' is available to convert all backward PERFORM names to a forward highlight of **RED** color, allowing the user to restart

forward tracing at a new PERFORMED routine, will turn off **RED** color to the just shown colors or highlighting for continued backwards tracing

Perform Warnings - 2nd Tab

Description of Warnings and Errors

One of the premises of Cool Auditing Tool is that most, if not all, of the COBOL program is organized under the control of the COBOL PERFORM verb. As mentioned earlier, the COBOL SORT verb with clauses INPUT PROCEDURE and OUTPUT PROCEDURE are treated as COBOL PERFORMs which, in essence, they are.

A COBOL program written in a manner to be controlled by GO TO s without the use of COBOL PERFORMs or the use of the older ALTER verb, will greatly distract from following COBOL logic with the use of the CAT software.

Multiple ENTRY points into a program, such as the use of ERROR routines with the use of CICS or with the use of home-grown Data Bases or software like SQL or DL1 need to be followed cautiously when following the tracing of COBOL PERFORMs. If the program is well designed and the COBOL technician is reasonably competent, then there should be less cause for concern in using CAT (Cool Auditing Tool) for COBOL.

When the use of COBOL PERFORMs is well established, there is still always concern that a PERFORM error exists that can throw off the logic of a COBOL program.

It is expected that an experienced COBOL technician has some understanding of what a COBOL PERFORM error is and how logic can be destroyed. Just the same, two brief examples are given here:

- A GO TO within a PERFORMed routine jumps outside of a COBOL range and rather than branching to an End of Job routine, lets processing continue and branching to the end of the original PERFORMed routine never happens or happens at a different sequence of logic than is expected for program processing.
- A COBOL technician copies a grouping of PERFORMed code and duplicates the code elsewhere in the program, renaming the paragraph names, but forgets to rename a used GO TO that branches to the routine exit as shown next:

Original code: C-USER-FUNCTION.
 MOVE ..
 IF ..
 GO TO C-EXIT.
 C-EXIT. EXIT.

Duplicated code: D-USER-FUNCTION.
 MOVE ..
 IF ..
 GO TO C-EXIT.
 D-EXIT. EXIT.

- Executing the new D-USER-FUNCTION and further executing a condition within that routine causing a branch to C-EXIT may create a very undesirable error in program and sometimes these conditions can exist in program logic for a long time without being identified.

COBOL Perform Warnings Window

COBOL PERFORM errors are not limited to the examples given below.

Quick View	Perform Warnings	Cobol Errors	Options	Narrative Selection	Debug
PERFORM Warnings					
Count	Type & Seq Nbr(s)				
01	GO TOs leaving the range of a PERFORM				
	0528				
01	Backward GO TO's				
	0436				
Unused Procedure Names					
	Seq#	Section/Paragraph Name	Unused	SECTIONS	
	538	G-LEAVE-FOR-SORT-IP-AND-OP		Message or Warning	
				SECTION not externally referenced	
			Unused	PARAGRAPHS	
	Seq#	Section/Paragraph Name		Message or Warning	
	539	GO10-DO-SORT		PARAGRAPH not externally referenced	
Normal PERFORM activity					
Count	Type & Seq Nbr(s)				
19	PERFORM SECTIONS				
	0192	0193	0203	0214	0217
	0219	0299	0334	0339	0344
	0360	0374	0375	0401	0406
	0411	0441	0448	0581	
05	PERFORM Paragraphs				
	0516	0517	0522	0543	0544
08	GO TO Paragraphs				
	0436	0502	0508	0512	0518
	0524	0528	0545		
PERFORM Warnings & activity Not found:					
GO TOs out of SORT INPUT or OUTPUT PROCEDURE					
PERFORM out of SORT INPUT or OUTPUT PROCEDURE					
Backward PERFORMs					
GO TO to a SECTION					

Four types of Information

To assist in eliminating PERFORM errors, this PERFORM Warning Tab is created with four categories of reporting as described below. Each category is highlighted in color. The listing of each of these four categories may be turned on or off from within the 'Options' Tab.

. PERFORM Warnings

Several types of warnings may be reported on here, but are only reported if one or more warnings are found. See above where there are two examples of PERFORM Warnings along with a count of each type of error.

. Unused Procedure Names (Dead Code)

This category is not truly a Perform Warning, but instead a list of potentially dead code or unused SECTIONs or PARAGRAPHS found, where they are found and any other appropriate comments. See the example on the previous page showing one SECTION not accessed and one PARAGRAPH not accessed.

. Normal PERFORM activity

Normal PERFORM activity is categorized as PERFORMs of Paragraphs and Sections and also GO TO s that reach a point with the range of a PERFORM of a Paragraph or Section. See example on previous page.

. PERFORM Warnings & activity NOT found:

This routine lists a grouping of potential Perform Warnings in each category that were not found withing this program.

COBOL Errors - 3rd Tab

Quick View	Perform Warnings	Cobol Errors	Options	Narrative Selection	Debug
Moderate Perform Warning - Backward GO TO at sequence number 000436					
Major Perform Error - GO TO leaves PERFORM Range at sequence number 000528					
SQ-NBR	MSG-NBR	DESCRIPTION			
00272	DCDM5J01-W	THE NAME (INV-M-3RD-DAT) DOES NOT MATCH TO A DATA DIVISION NAME			
TOTAL NUMBER OF W LEVEL MESSAGES = 1					
TOTAL NUMBER OF MESSAGES = 1					

COBOL Errors Sample

Errors in scanning the COBOL program and its associated COPY members are listed in this section. The message includes a sequence number of where in the program the error occurs, a message number in case there are further questions about this error and a message description describing what happened to cause this error. See example shown on previous page.

Also, two types of errors, if either is found, will be reported ahead of the SYNTAX errors.. First, if an unprintable character is found in the source listing, an error message will report that here.

Second, if Major (or by selection from 'Options' tab, Moderate or Minor) Perform Warnings are found they will be shown here. See example on previous page for both a Moderate and a Major Perform Error being listed.

Note – These PERFORM warnings are also reported under Perform Warnings Tab. As not all programmers will always visit the PERFORM Warnings tab, they will be shown here according to a ranking of Major, Moderate, or Minor. Selection may be used in 'Options' tab to control exactly which categories are listed here.

A count is given for all SYNTAX errors, but this count will not include any errors listed at the top of the report used for unprintable characters or listed PERFORM Warnings and before the heading shown just below:

SQ - NBR	MSG - NBR	DESCRIPTION
----------	-----------	-------------

Narrative Display

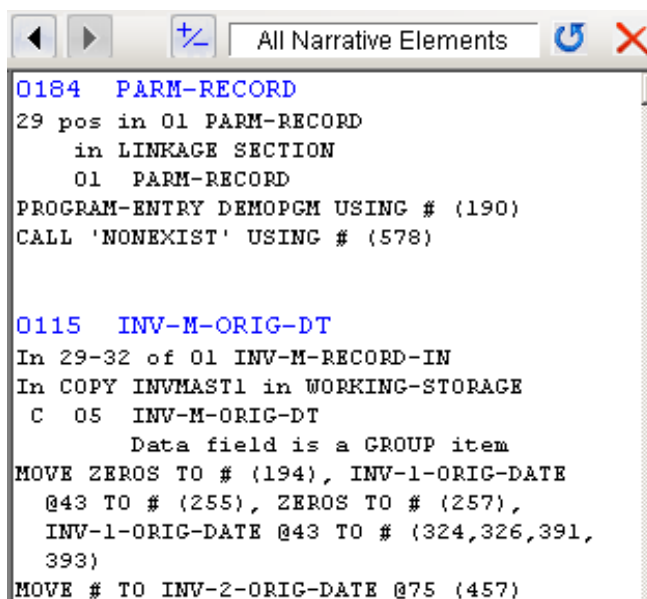
The Narrative Window in CAT is used to provide full information (both attributes and references) on selected data names.

To choose a data name for selection, simply click on a data-name (not paragraph name) anywhere in the source listing window. Data-names are highlighted (normally in **blue**) in the Data Division and also in the Procedure Division.

One of the aims of CAT is to keep the programmer/technician or COBOL auditor within the Procedure Division as much as possible and one of the ways we do this is to allow a data-name to be quickly viewed with as much information on the data-name as the user wants to see.

Using the +/- Button

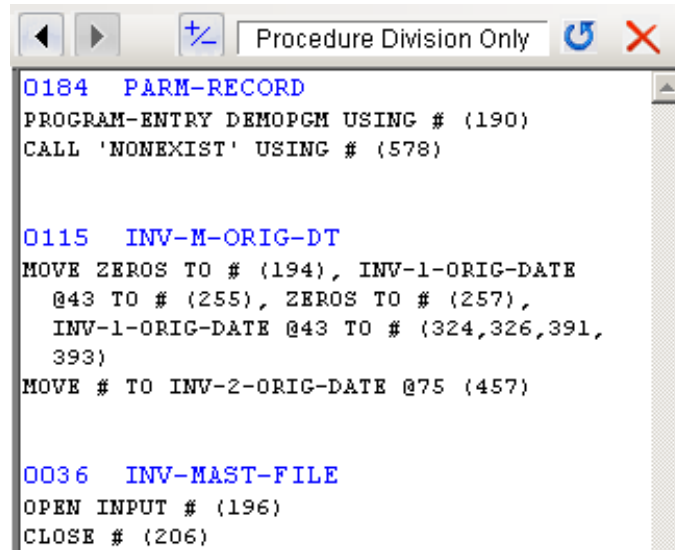
Two types of information or documentation are provided in the Narrative Window for each selected data-name and these two types are Data Division information and Procedure Division references. Clicking back and forth on the +/- button shown in the top line of either example shown below show two formats for narrative, one with both Data Division and Procedure Division together and the other for only Procedure Division with Data Division attributes omitted.



```
0184  PARM-RECORD
29 pos in 01 PARM-RECORD
    in LINKAGE SECTION
    01  PARM-RECORD
PROGRAM-ENTRY DEMOPGM USING # (190)
CALL 'NONEXIST' USING # (578)

0115  INV-M-ORIG-DT
In 29-32 of 01 INV-M-RECORD-IN
In COPY INVMAS1 in WORKING-STORAGE
C 05  INV-M-ORIG-DT
    Data field is a GROUP item
MOVE ZEROS TO # (194), INV-1-ORIG-DATE
    @43 TO # (255), ZEROS TO # (257),
    INV-1-ORIG-DATE @43 TO # (324,326,391,
    393)
MOVE # TO INV-2-ORIG-DATE @75 (457)
```

All Narrative Elements



```
0184  PARM-RECORD
PROGRAM-ENTRY DEMOPGM USING # (190)
CALL 'NONEEXIST' USING # (578)

0115  INV-M-ORIG-DT
MOVE ZEROS TO # (194), INV-1-ORIG-DATE
    @43 TO # (255), ZEROS TO # (257),
    INV-1-ORIG-DATE @43 TO # (324,326,391,
    393)
MOVE # TO INV-2-ORIG-DATE @75 (457)

0036  INV-MAST-FILE
OPEN INPUT # (196)
CLOSE # (206)
```

Procedure Division Only

Individual options within the 'Narrative Selection' Tab allow the user to specify how much Data Division or how much or what type of format is used in showing Procedure Division references. Also for Procedure Division references, a special narrative condensed format is used to list each found Procedure Division COBOL statement. *(For instance, a # is used to indicate the data-name being referenced to save space in the Narrative window.)*

Besides these Individual options under the 'Narrative Selection' Tab, there is a selection button (+/-) on top of the Narrative window that may be used at any time to go back and forth between just shown Procedure Division references OR also to show as much Data Division information as indicated in the options, shown underneath the 'Narrative Selection' Tab.

COBOL Narrative (two types)

Narrative is created in the Narrative Window (on the right side) when a user clicks on a data-field link from the COBOL Source Listing (including COPY members) on the top left side window of CAT (Cool Auditing Tool). Two basic types of COBOL Narrative are available with the COBOL Narrative window after a user clicks on a data-name in the COBOL source listing on the left side of CAT.

The first type is DATA DIVISION Narrative which includes all sorts of information from the Data Division. Different types of information available here are discussed just below under the DATA DIVISION NARRATIVE heading.

The second type is PROCEDURE DIVISION NARRATIVE which includes both Direct and Indirect Procedure Division usage of the data-field selected on.

- Direct Narrative lines are a condensed format of all Procedure Division statements which directly use or reference this data name. There are two sequences that this narrative is available in and this will be discussed under the heading PROCEDURE DIVISION NARRATIVE.

Indirect References are references to Procedure Division statements which indirectly reference the selected data field, such as referencing a REDEFINES field to the data name selected or referencing a GROUP field that holds the data name selected or other situations where a field may be referenced to a field with overlapping field positions.

DATA DIVISION NARRATIVE

From-To Positions relative to the 01 record that the field belongs to

For an 01 record, the total size of the record will be listed along with the 01 record in a format similar to the following:

120 pos in 01 INV-1-RECORD-IN

For other fields residing within an 01 record, both a From and To position are shown in a format similar to the following:

In 29-30 of 01 INV-2-RECORD-IN

Level Number of the Data Field

For 01-49, 66, 77, or 88 level entries, the level number is shown ahead of the data-name in a format similar to the following:

05 INV-2-ORIG-DATE

Within a COPY Member

Two forms of documentation are available for a data field belonging to a COPY member.

The first format puts the letter C ahead of the level number and data name similar to the following:

C 05 INV-2-ORIG-DATE

The second format provides a line of documentation showing the COPY name and the SECTION that the 01 record or field belongs to in a format similar to the following:

In COPY **MembName** in WORKING-STORAGE

where MembName is the COPY Member name.

SECTION Name that the Data Field comes from

Section Name takes up a line by itself in the documentation. However, if the field is within a COPY member, then that also will be shown on the same line as in the previous example under 'Within a COPY Member'. Without COPY member, a format similar to the following is used:

in FILE SECTION

or

in WORKING-STORAGE

EXPANDED SEQUENCE NUMBER of where the Data Field resides

The Expanded Sequence Number of where the Data Field resides in the CAT COBOL listing on the left side is put with Data Field Name on one line in a format similar to the following:

0115 **INV-M-ORIG-DT**

and this line is highlighted in color and listed first before any other Narrative.

Notes

- 1. This line is always shown even if other Data Division Narrative is omitted with use of Tab button just above the Narrative Window.*
- 2. Expanded Sequence Number refers to a complete sequence number in the COBOL source listing on the window to the left of the narrative window which contains included COPY members and any INCLUDES.*

PIC, USAGE, and VALUE

These clauses are listed along with the level number and data name but directly underneath the level number and data name line in a format similar to the following:

```
10    INV-2-BASIC-COST
      Pic  S9(5)V99    Usage  COMP-3
      Value  ZEROS
```

GROUP Fields

By option, GROUP fields may be documented in a format similar to the following:

```
Data field is a GROUP item
```

DECIMAL DIGITS present

By option, fields with Decimal Digits may be identified in a format similar to the following:

```
Numeric Field had 02 DECIMAL digits
```

REDEFINES listed

By option, REDEFINES are listed when a REDEFINES is present for this data field and shown in a format similar to the following:

```
05    WS-ACCT-FIELD-2
      REDEFINES  WS-ACCT-FIELD
```

88 LEVEL ENTRIES

88 Level Entries are shown in a format that mimics what an 88 level looks like in the source listing in a format similar to the following:

```
88    INV-1-ACCT-BETA    VALUE  004.
```

The line will be either on one line or more than one line depending on the space available.

DATA NAME

The Data Name is listed in many formats and always on the first line that is highlighted.

FILE NAME

Work is currently being done to list File Name with the 01 record that it belongs to and other information such as RECORD CONTAINS and BLOCK CONTAINS.

OCCURS

Work is being done to list the presence of an OCCURS clause for the name where the OCCURS exists.

PROCEDURE DIVISION NARRATIVE

Direct Narrative refers to direct use of a data-name within the PROCEDURE DIVISION.

Two examples of COBOL lines that create DIRECT use for a field TR-INVOICE-YEAR follow:

```
MOVE    2015    TO    TR-INVOICE-YEAR.  
IF TR-INVOICE-YEAR > 2013,
```

To better clarify DIRECT referencing, an example of an INDIRECT reference is given here. In the just below MOVE, TR-INVOICE-DATE that belongs to record TR-TRANS-RECORD is modified indirectly with:

```
MOVE SPACES TO TR-TRANS-RECORD.
```

Indirect (implied) references to fields like TR-INVOICE-DATE will be handled later under heading **INDIRECT NARRATIVE**.

Direct Narrative in CAT refers to special condensed formatting of PROCEDURE DIVISION statements that reference DATA DIVISION names and are then condensed to closely resemble the original PROCEDURE DIVISION statement or multiple PROCEDURE DIVISION statements in a format for instant readability.

This is what happens in creating the PROCEDURE DIVISION narrative:

1. A condensed format takes up less SPACE and makes fitting the narrative into the 43 characters of the Narrative Window much easier
2. The data name referenced continually in one set of narrative is only listed once at the top of the narrative and the character # is used in place of the data-name to save space in the narrative as shown below:

```
MOVE ZEROS TO #
```

Several Procedure Division Statements may be correlated together showing multiple sequence numbers to indicate they exist and show where they reside as shown below:

```
ADD 1 TO # (1045,1403,1732)
```

When more than one data name is referenced in the field, a pointer can point to where the other data name resides in the expanded source listing for clarification as shown below:

```
MOVE # TO OTHER-FIELD @255
```

Two different SORT sequences are available for displaying the narrative lines as follows:

- Sort by Verb Name (such as ADD, IF, MOVE) and then by sequence number
- Sort by sequence number in the order the statements appear in the program

The first format is more condensed, and is sometimes preferred as it may take less space.

The second format is less condensed, and is sometimes preferred as it shows a clear order of when PROCEDURE DIVISION statements are found.

Narrative Examples:

Consider a PROCEDURE DIVISION statement similar to the following:

```
ADD 1 TO WS-NEXT-COLUMN
```

at sequence number 1019 in the source listing.

Then, when the user clicks on WS-NEXT-COLUMN from anywhere in the source listing, a narrative entry is created for WS-NEXT-COLUMN showing where it exists in the DATA DIVISION and also creates a condensed PROCEDURE DIVISION statement in the narrative window similar to the following:

```
0349 WS-NEXT-COLUMN  
Add 1 to # (1019)
```

where the number 1019 in parenthesis indicates where in the PROCEDURE DIVISION this ADD exists.

If a second and third identical ADD also existed at lines 1145 and 1382 in the PROCEDURE DIVISION, then the narrative would be expanded to look like the following:

```
0349  WS-NEXT-COLUMN
      Add 1 to # (1019,1145,1382)
```

If another data name is involved in the associated Procedure Division statement, then the sequence number of that data name is given following an @ character. Consider a PROCEDURE DIVISION statement at sequence number 1005 as follows:

```
IF  WS-NEXT-COLUMN = WS-MAX-COLUMN
```

Then the following line of narrative would be added to the narrative in one of two orders depending on the SORT sequence specified under Narrative Options.

For first format, the following lines would appear:

```
0349  WS-NEXT-COLUMN
      Add 1 to  # (1019,1145,1382)
      IF # = WS-MAX-COLUMN @321 (1005)
```

For second format, the following lines would appear in slightly altered SORT sequence:

```
0349  WS-NEXT-COLUMN
      IF # = WS-MAX-COLUMN @321 (1005)
      Add 1 to  # (1019,1145,1382)
```

The reason for this different sequence is that in the second sequence, Procedure Division statements are sorted in ASCENDING order, while in the first sequence, they are sorted in VERB order.

Consider, when an extra PROCEDURE DIVISION line is added for this IF at sequence number 1300, then the following narrative sequences would be added for first format as follows:

```
0349  WS-NEXT-COLUMN
      Add 1 to # (1019,1145,1382)
      IF # = WS-MAX-COLUMN @321 (1005,1300)
```

And for the second option the following SORT sequence would be as follows:

```
0349  WS-NEXT-COLUMN
IF # = WS-MAX-COLUMN @321 (1005)
Add 1 to # (1019,1145)
IF # = WS-MAX-COLUMN @321 (1300)
Add 1 to # (1345)
```

Notice here, that the Narrative is two lines longer and is such because the PROCEDURE DIVISION statements are displayed in the order they appear in the program.

Sometimes only part of a PROCEDURE DIVISION statement is shown. Consider a PERFORM statement at line 951 accessing the name WS-NUMBER-TIMES of the program as follows:

```
PERFORM  S-ACCT-TOTAL  THRU  S-ACCT-EXIT
        UNTIL  WS-NUMBER-TIMES = ZERO
```

The Narrative produced would look like the following:

```
Perform-Until # = ZERO (951)
```

In most instances, however, the entire COBOL statement from the Procedure Division is shown as for the following SUBTRACT statement at line 1041 in source listing:

```
SUBTRACT  1  FROM  WS-NUMBER-TIMES
```

The Narrative produced would look as follows:

```
0252  WS-NUMBER-TIMES
Subtract 1 from #(1041)
```

In many cases, where the Procedure Division statements are similar, but relate to different operands, the narrative will not reprint the left most part of the Procedure Division statement that is unchanged between the similar statements. For example:

```
If # = 3 (942,965,1022), 4 (945,968,1025)
Move 03 to # (497), 04 to # (501)
```

The above narrative indicates there are 3 PROCEDURE DIVISION statements as follows:

```
IF  WS-MAX-PD-DIGITS = 3
```

and 3 PROCEDURE DIVISION statements as follows

```
IF WS-MAX-PD-DIGITS = 4
```

and one statement

```
MOVE 03 TO WS-MAX-PD-DIGITS
```

and one statement

```
MOVE 04 TO WS-MAX-PD-DIGITS
```

Note, the above example given earlier for Sort Format 1. If SORT format 2 were used the narrative would be quite expanded and looks as follows:

```
MOVE 03 TO # (497)
MOVE 04 TO # (501)
IF # = 3 (942)
IF # = 4 (945)
IF # = 3 (965)
IF # = 4 (968)
IF # = 3 (1022)
IF # = 4 (1025)
```

When no narrative appears for a Data Division name, this indicates that the data name is unreferenced.

Special Note - There is a **third** Type of Direct Narrative used outside the Narrative Window

Regardless of the SORT sequence used to condense the narrative here, there is an option which allows the Procedure Division Narrative to be shown line by line in the **Quick View Window** underneath the appropriate PERFORMed Paragraph or Section where the Procedure Division Verb occurs.

When this option is turned on and invoked the formatting will not be condensed but will be expanded to look like the original PROCEDURE DIVISION statement except that also an @ character followed by the sequence number of this Data Division Name will be inserted after the Data Name in the **Quick View Window** as shown in the below example:

```
Move WS-HOLD-DT to INV-M-ORIG-DT @115
```

Not only does the @115 point to where INV-M-ORIG-DT resides, it also indicates that this is the field in the MOVE, or other verb, that was selected in the Narrative Window by the user.

INDIRECT REFERENCES

If the Narrative Selection option is turned on to show Indirect References, then a condensed summary of Indirect References will be shown beneath the Direct Narrative. Indirect Narrative always begins with the word **Indirectly** and is followed with one of three words (**Changed**, **Used**, or **Tested**) and by an @ character and one or more sequence numbers separated by commas.

These sequence numbers indicate there is one or more indirect references at the sequence number or numbers that follow the @ character. After the last sequence number, there may be another occurrence of the three words (**Changed**, **Used**, or **Tested**) with more sequence numbers as follows:

Indirectly Tested @57, Used @40,57

When narrative is present, the Indirect References will be shown after the Direct Narrative for the data field.

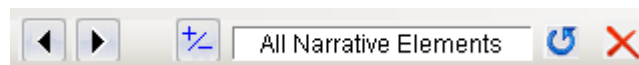
The word **Changed** indicates this data-field is in some way indirectly changed or modified. The word **Used** indicates that this data-field was used to indirectly modify or change another field. The word **Tested** indicates that this data-field was used to indirectly compare another field.

CAT Navigation Button Operation

There are two sets of window navigation buttons. Each set contains left and right arrow buttons. One set is associated with the Narrative display and one is for the Quick View window display.

Narrative Navigation Operation

The arrow navigation buttons above the Narrative display are used to navigate to previous or next narrative data that has been displayed by clicking on the 87 Character Display. These buttons only affect the narrative window display.



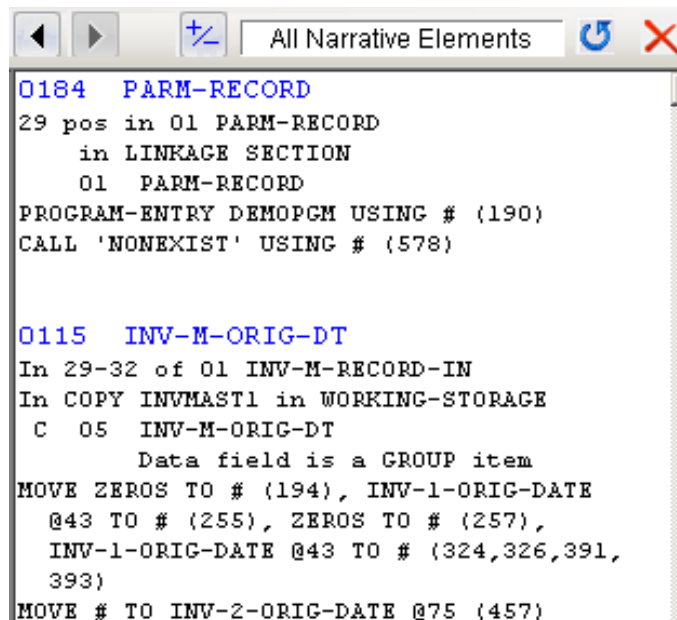
Narrative upper control block section

The arrow navigation buttons will only be enabled when the Narrative window has data sets that can be navigated to. For instance, when the placement of the first Narrative data element is added to the display, both of the arrow buttons will be disabled. When the second data element is added to the display, the left arrow will be enabled and will stay enabled as additional elements are added. If the left arrow button is clicked, the Narrative window will display the previous narrative data set and the right arrow button will be enabled. Whenever the last Narrative data set is displayed the right navigation arrow button is disabled.

The red **X** Clear button is used to clear the current contents of the Narrative display

The +/- button will select between the display of all narrative element information or narrative information. Two images below show the two format available from using the +/- toggle button. The first shows 'All Narrative Elements' including from both the Data and Procedure Division. The second shows 'Procedure Division Only'.

1.

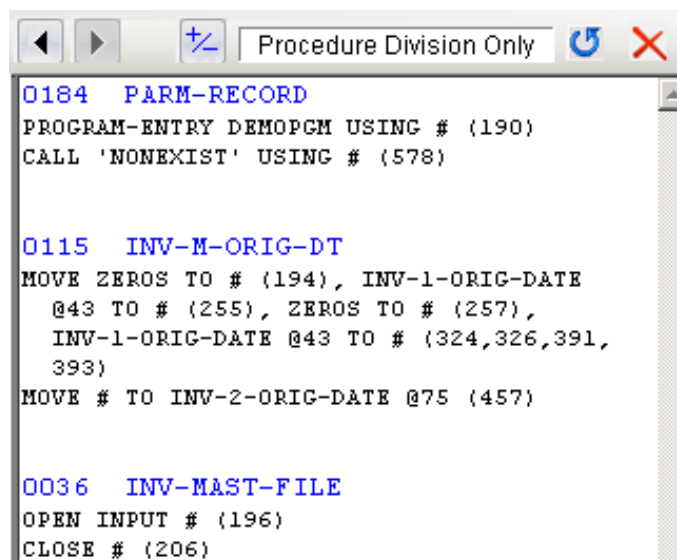


A screenshot of a window titled "All Narrative Elements". The window contains assembly code for two sections: "0184 PARM-RECORD" and "0115 INV-M-ORIG-DT". The code for "0184 PARM-RECORD" includes a position indicator, linkage section, and program entry. The code for "0115 INV-M-ORIG-DT" includes a position indicator, data field description, and several move instructions.

```
0184 PARM-RECORD
29 pos in 01 PARM-RECORD
    in LINKAGE SECTION
    01 PARM-RECORD
PROGRAM-ENTRY DEMOPGM USING # (190)
CALL 'NONEXIST' USING # (578)

0115 INV-M-ORIG-DT
In 29-32 of 01 INV-M-RECORD-IN
In COPY INVMAS1 in WORKING-STORAGE
C 05 INV-M-ORIG-DT
    Data field is a GROUP item
MOVE ZEROS TO # (194), INV-1-ORIG-DATE
    @43 TO # (255), ZEROS TO # (257),
    INV-1-ORIG-DATE @43 TO # (324,326,391,
    393)
MOVE # TO INV-2-ORIG-DATE @75 (457)
```

2.



A screenshot of a window titled "Procedure Division Only". The window contains assembly code for three sections: "0184 PARM-RECORD", "0115 INV-M-ORIG-DT", and "0036 INV-MAST-FILE". The code for "0184 PARM-RECORD" includes a position indicator and program entry. The code for "0115 INV-M-ORIG-DT" includes a position indicator and several move instructions. The code for "0036 INV-MAST-FILE" includes a position indicator and open/close instructions.

```
0184 PARM-RECORD
PROGRAM-ENTRY DEMOPGM USING # (190)
CALL 'NONEXIST' USING # (578)

0115 INV-M-ORIG-DT
MOVE ZEROS TO # (194), INV-1-ORIG-DATE
    @43 TO # (255), ZEROS TO # (257),
    INV-1-ORIG-DATE @43 TO # (324,326,391,
    393)
MOVE # TO INV-2-ORIG-DATE @75 (457)

0036 INV-MAST-FILE
OPEN INPUT # (196)
CLOSE # (206)
```

Narrative +/- toggle options

Quick View Navigation Operation

The following display shows ways to navigate within the Quick View Window.



QVW Navigation Buttons

The set of navigation buttons (**green**) will move the Quick View Window to previous or next sets of data that have resulted from the selection of data from the 87 Character Window and from Narrative added to the Quick View Window from the Narrative Window.

The top row of buttons includes a blue **Refresh** button and a red **X** Clear button. The Refresh button will redisplay the Quick View window using the current contents of the Narrative window. This allows new narrative elements just added to or removed from the Narrative Window to be added or removed to the Quick View Window as the Quick View Window is redisplayed.

The **red** Clear button is used to clear the current contents of the Quick View Window display.

The Quick View Window navigation buttons will only be enabled when the Quick View Window has previous Quick View Window views that can be navigated to. For instance, when the first Quick View Window is displayed following the running of a source COBOL file, both of the arrow buttons will be disabled. When the second view is added to the display, the left arrow will be enabled and will stay enabled as additional views are added. If the left arrow button is clicked, the Quick View window will display the previous Quick View view and the right arrow button will be enabled. Whenever the last Quick View view is displayed the right navigation arrow button is disabled.

Modify Program Settings

Sometimes an option or two needs to be changed for processing a COBOL program, such as the QUOTE option that may be used to indicate that the program uses a double quote rather than a single apostrophe for scanning and identifying literals within VALUE clauses and in the Procedure Division within a COBOL program.

Changing these options is done with check boxes by turning an option on or off. These options are set (and may be reset) to established defaults by using a check box marked 'Reset...' near the bottom and to the right of this window.

Changing the line count for number of lines is done by checking the 'Change Line Count' check box near the bottom and on the left side, and then by readjusting the numeric field in the associated line count box near it. This changes the line count used in the 'Alternate Compile Listing' produced in scanning the program and does not affect the other CAT windows produced.

Below is a picture of the window used to specify options followed by a description of each option:

Alternate Compile Listing - Modify Program Settings

Other Options

☒ ASKIP	Add Seq Nbrs for SKIP1, SKIP2, & SKIP3	☐ IEJECT	Ignore EJECTs when creating SOURCE Listing
☐ C68	Treat COBOL program as 1968 or earlier	☐ NIS	Use Sequence Nbr as primary SORT field in Narrative
☒ DNF	Print Warning Messages for missing names	☐ NRSPACE	Insert blank lines to separate Narrative between names
☐ EIB	Insert CICS COPY member DFHEIBLK	☐ NUC	Force all Narrative to be UPPER Case
☒ ERRORS	Print ERROR Messages	☐ QUOTE	Literals use double quotes ("), not single
☒ FTB	Put FROM-TO positions in 73-80; if 73-80 = SPACES	☒ RESOLVE	Resolve COPY members - Not using may produce errors
☐ FTO	Force FROM-TO pos (relative to 01 recd) in cols 73-80	☒ UNREF	Include UNREFERENCED data names in cross reference
☐ F16	Put FROM position (relative to 01 record) in cols 1-6	☐ VR3	Emulate 1974 Version 3 and earlier COBOL
☐ HFT	Build FROM-TO positions in hexadecimal, not decimal	☒ VS2	Use for 1985 or newer COBOL
☐ IDD	Include DATA DIVISION Literals in LITERALS report		

☒ Change page line count

☐ Reset this Window's options to established defaults

Additional Options Provided By Marble Customer Support

Save **Back**

Program Options

The button on the bottom of the image just shown labeled 'Additional Options Provided by Marble Customer Support' is not for use except in situations where customer support is being provided and then needed only as directed by Marble Personnel.

When options have been viewed without any changes OR changes have been made, use the *Save* or *Cancel* button to exit. If changes have been made that the user wants to keep, use the *Save* button.

The following is a description of what each option accomplishes by turning the option on or off.

ASKIP	This option processes IBM compiler directing commands (SKIP1, SKIP2, and SKIP3) for inserting 1, 2, or 3 blanks lines respectively within the generated Alternate Compile Listing report. This option is defaulted to ON and usually will not cause any problems left on for non-IBM COBOL.
C68	Treat this program as a 1968 ANSI standard program. Only use this option if the user knows the program is very old and conforming to 1968 standards. This option is defaulted OFF.
DNF	If a data field is used within CAT that is not defined within the Data Division in the program or defined within a COPY member that is brought into the program, the data field will be documented as missing in an error message. This option is defaulted to ON. Turning the option OFF (not usually recommended) will make these warning messages disappear.
EIB	CICS programs typically use this COPY and if it is not otherwise brought in, it may be included within the program by checking this option ON. The default for this options is OFF.
ERROR	This option is defaulted to ON. It needs to be left on for use in CAT. It controls the printing of ERRORS during scanning the COBOL program and associated COPY members.
FTB	This option is defaulted to ON. It indicates in the Data Division, the From-To positions of each data field relevant to the 01 record it belongs to, and will be inserted into columns 73-80 of the generated Alternate Compile Listing report <u>only</u> if columns 73-80 are all spaces.
FTO	This option forces From-To positions into columns 73-80 of the generated Alternate Compile Listing report.
F16	This option will place the From-To positions into columns 1-6 of the generated Alternate Compile Listing report.

HFT	Build all From-To positions into Hexadecimal numbering, not Decimal format with the generated Alternate Compile Listing report.
IDD	Include Data Division Literals in the LITERALS report of the generated Alternate Compile Listing report. If this option is not checked, <u>only</u> Procedure Division Literals will show in Literals Report.
IEJECT	Ignore EJECTs when creating SOURCE listing of the generated Alternate Compile Listing report. This option is defaulted to OFF. If EJECTs are not present, then this option is ignored.
NIS	This option is not available for source listing of the generated Alternate Compile Listing Report. It is available in CAT options with different wording to provide two formats for COBOL Narrative.
NRSPACE	Insert a blank line to separate Narrative between names in the generated Alternate Compile Listing report. This option is defaulted to OFF and not especially recommended for turning ON for source listing.
NUC	Force all Narrative to be UPPER case. This option is defaulted to OFF.
QUOTE	Use when Literals have double quotes ("... ") rather than single apostrophes for enclosing each literal.
RESOLVE	This option should always be ON. It is used to state that COPY members should be resolved. Turning this option OFF may produce many errors.
UNREF	This option is defaulted to ON. It indicates that unreferenced names are to be included in the Data Division Cross Reference Report in the generated Alternate Compile Listing report.
VR3	This option is defaulted to OFF. When used, it tells scanning to look for 1974 Version 3 and earlier COBOL.
VS2	This option is defaulted to ON. This option is needed for ANSI standard 1985 COBOL and newer COBOL versions.

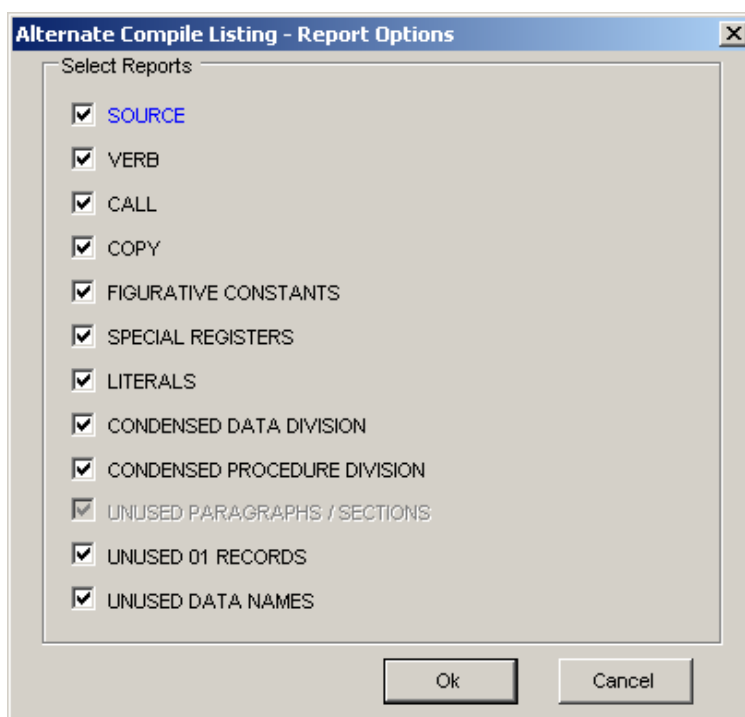
Modify Report Options

This panel, shown below, allows the user to select which reports will show in generated 'Alternate Compile Listing' (Source Listing produced by CAT, but separate from normal CAT Windows). The Alternate Compile Listing is similar to what is produced for mainframe use with Marble software DCD III. It is made available with CAT and sometimes maybe a very useful report.

It is recommended that all report options be left on. The biggest report is the Source Listing (85% or more of listing will be the Source Listing). The other reports are typically quite small.

All reports are defaulted ON. The only report that must stay on for CAT use is the Unused Paragraphs / Sections report. It is also recommended that the source listing be left on.

Below is a picture of the SELECTION window for Reports. Following that is a brief description of each report. Actually looking at the 'Alternate Compile Listing' will give an even better view and understanding of each report, especially the Source Listing with its unique narrative.



Report Options

SOURCE Report

This report is the entire source listing with COPY members expanded and with COBOL Narrative shown for each data field in the Data Division. It also shows references in the Procedure Division, both pointing back to the Data Division to view COBOL Narrative and with Transfer of Control Narrative showing where to go to follow PERFORMs and GO TO s, along with a range for each PERFORM. It is usually recommended that this report be left on.

VERB Report

This report shows all Procedure Division Verbs and where they are used in the Procedure Division.

CALL Report

This report shows where all CALLs are used in the program.

COPY Report

This report shows where all COPY members are found.

FIGURATIVE CONSTANTS Report

This report shows all use of Figurative Constants (ZEROS, SPACES, LOW-VALUES, HIGH-VALUES, QUOTES) within the program.

SPECIAL REGISTERS Report

This report shows where SPECIAL REGISTERS, such as the use of ADDRESS OF or LENGTH OF, are in the program.

LITERALS Report

This report shows where literals are used in the Procedure Division. To include literals in the Data Division, see the heading 'Modify Program Settings'.

CONDENSED DATA DIVISION Report

The report for Data Division References in the 'Alternate Compile Listing' is very small because a detailed listing of all References is shown in COBOL DATA DIVISION NARRATIVE produced by CAT. Data Names are listed in 3 columns with a reference back to the Data Division where all references are shown.

CONDENSED PROCEDURE DIVISION Report

The report for Procedure Division References in the 'Alternate Compile Listing', like the Data Division Condensed Report, is also very small with one reference back to a paragraph or section allowing the user to get all references shown at that point in the Procedure Division.

UNUSED PARAGRAPHS / SECTIONS Report

This report is required on for CAT and it shows all possible dead code OR Unused Paragraphs and Unused Sections with added information such as ENTRY points within that code.

UNUSED 01 RECORDS Report

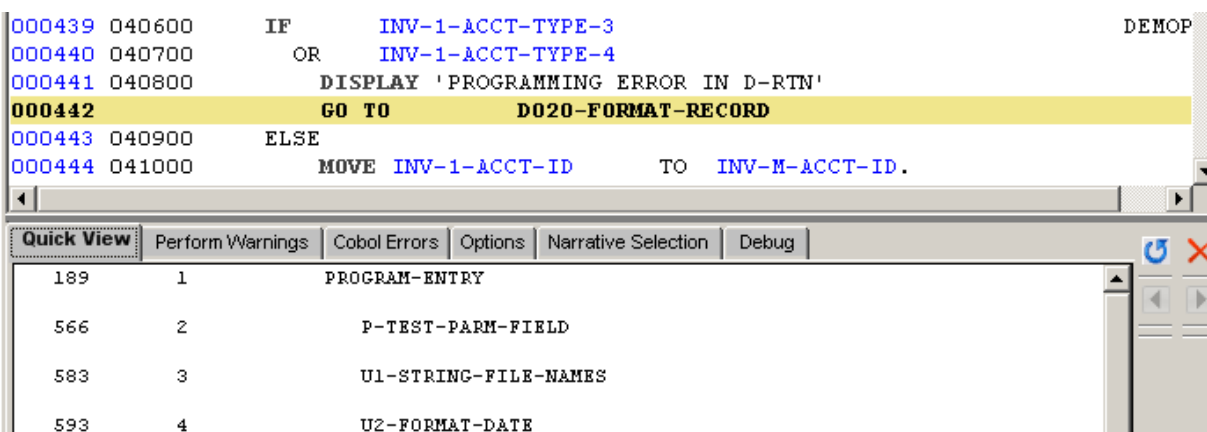
This report shows all 01 records that are un-referenced in this program.

UNUSED DATA NAMES Report

This report shows all un-referenced data names in the program with the exception that it will not show un-referenced data names that are present in a COPY member.

Tab Options Selection

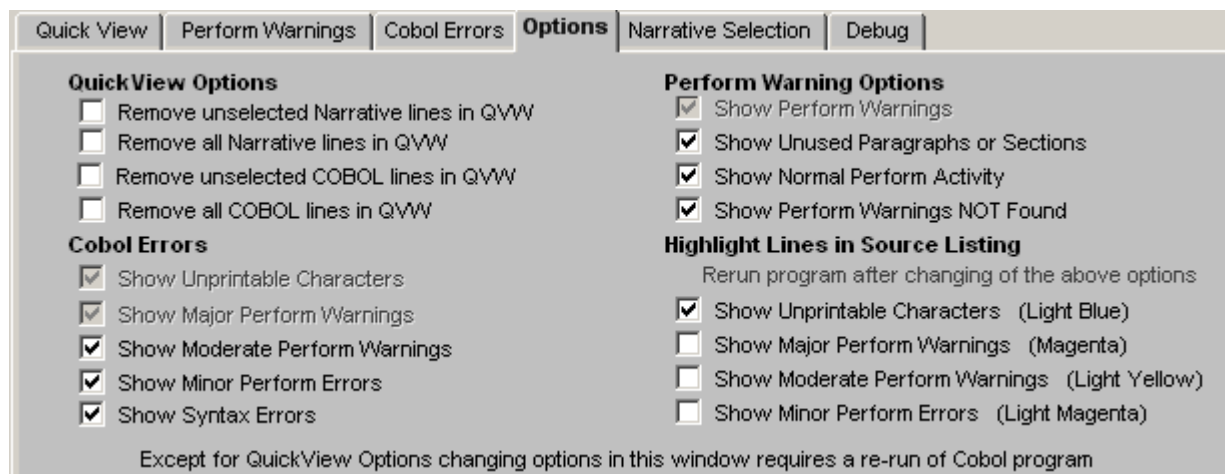
Located in the lower Tab Options area are two panes named Options and Narrative Selection. These two tab panes contain a number of option selection checkboxes that can be used to customize how information is displayed to a CAT user.



Viewing the Lower Tab Section

Options - 4th Tab

The Options Selection contain four different sections including options. One is for the Quick View Window, one is for Perform Warnings, and one for Cobol Errors and one for Highlighting certain error lines in the Source Listing. See 'Except for..' note shown at bottom of this picture.



Options Tab Panel

The Quick View Options tab allow the user to include or exclude specific sets of information, including selected or unselected COBOL and CAT narrative lines. The Perform Warning Options pane includes one defaulted selection that can not be changed by the user, as well as three selectable options.

The Cobol Errors Options are divided into two sections, the first of which controls information that is displayed in the Cobol Errors pane and the second set of selections controls visual highlighting in the main 87 character display window.

Quick View Options Detailed Description

Remove unselected Narrative lines in QVW

Procedure Division Narrative shown within the Narrative window on the right side of CAT, such as the following narrative “Move ZEROS to # (205)”, is also generated (by option) for inclusion in the Quick View Window and is placed either at the end of the Quick View Window as an unselected Narrative Line OR underneath the relevant PERFORM in the Quick View Window (or relevant PERFORM's if there are two similar PERFORM's with overlapping ranges, both covering the range of this line, i.e. (205) in the “Move ZEROS..” example above). Checking this option allows the Narrative to print underneath the relevant PERFORM but will not list the narrative at the end if unselected under one or more PERFORM's. Also, see next option.

Remove all Narrative lines in QVW

Procedure Division Narrative shown within the Narrative window on the left side of CAT, such as the following narrative “Move ZEROS to # (205)”, is also generated for inclusion in the Quick View Window. Checking this option prevents this narrative from being included anywhere in the Quick View Window.

Remove unselected COBOL lines in QVW-(feature not available in this version)

Procedure Division code may be brought into the Quick View Window by selecting (clicking on) a Procedure Division Verb in the source listing (upper window on left side of CAT). When a Verb is selected, CAT will determine the begin and end range of the COBOL verb and have that code placed at the end of the Quick View Window as unselected COBOL line(s) OR underneath the relevant PERFORM in the Quick View Window (or relevant PERFORM's if there are two similar PERFORM's with overlapping ranges both covering the range of this line. Checking this option allows the COBOL line(s) to print underneath the relevant PERFORM but will not list the COBOL lines at the end if unselected under one or more PERFORM's.

Remove all COBOL lines in QVW**-(feature not available in this version)**

Procedure Division code is brought into the Quick View Window by selecting (clicking on) a Procedure Division Verb in the source listing (upper window on the left side of CAT). When a Verb is selected, CAT determines the begin and end range of the COBOL verb and places that code appropriately in the Quick View Window. Clicking on this option removes those lines from appearing anywhere within the Quick View Window.

Perform Warnings Options Detailed Description**Show Perform Warnings**

CAT uses logic to determine warnings within COBOL Performs. These warnings are grouped into categories, such as GO TO s leaving the range of a PERFORM, and backward GO TO s and then, in a report underneath the 'Perform Warnings' Tab, it shows a count for each category where there are one or more errors and lists the sequence number(s) that generated these warnings. This option is forced on, so that these warnings will print.

Show Unused Paragraphs or Sections

CAT uses tracing to show SECTIONS or PARAGRAPHS that can not be reached. When found, reporting will be grouped into separate reports for SECTIONS and for PARAGRAPHS and then shown within the 'Perform Warnings' Tab, after Perform Warnings are listed. This report will only be omitted if this option is not checked.

Show Normal Perform Activity

CAT uses an internal algorithm to show normal PERFORM behavior that will not lead to PERFORM errors in most circumstances. These include PERFORMs of PARAGRAPHS and of SECTIONS and also include GO TO s that do not leave a PERFORM range. Care must be taken to examine PERFORMs that have overlapping ranges, to look at ENTRIES into a program and CALLs leaving a program from a PERFORMed range and also to examine the PERFORM Warning reports shown first under this tab. The reporting of Normal Perform Activity may be turned off by un-checking or un-selecting this option.

Show Perform Warnings NOT found

For those categories used to determine Perform Warnings, such as GO TO s leaving the range of a PERFORM, and backward GO TOs. If there are no Perform Warnings in a particular category, these Categories will be listed last. To omit this report, simply uncheck or unselect this option.

COBOL Errors Detailed Description

Show Unprintable Characters

There are some ASCII characters that are unprintable in a Rich Text box (format used) for the COBOL source listing. If any of these characters are found, the line number in the source listing and beginning column are shown in an error message listed first underneath the COBOL Errors report.

These characters may be within a literal (Data Division VALUE, or Procedure Division literal) or within a COBOL comment where they may be a valid character or they may reside elsewhere in valid non-commented out or non-literal code where the unprintable character may be invalid. Turning this option on or off, controls whether or not these unprintable characters are shown. In this version of CAT, this option is forced on.

Show Major Perform Warnings

CAT uses an internal method to analyze PERFORMs and then categorize PERFORM Warnings according our own category of Major, Moderate, or Minor. An example of a Major Perform Error is a GO TO leaving the range of an established PERFORM. Turning this option on or off, controls showing these Major Perform Warnings. In this version of CAT, this option is forced on.

Note – Perform warnings are also shown under the tab 'Perform Warnings'. However, when shown in that tab, the warnings are NOT categorized as Major, Moderate, or Minor.

Show Moderate Perform Warnings

An example of a Moderate Perform Warning is a Backward GO TO, allowing a possible disruption of PERFORM depending on other logic within the COBOL program. Turning this option on or off controls showing these Moderate Perform Warnings

Note - Perform warnings are also shown under the tab 'Perform Warnings'. However, when shown in that tab, the warnings are NOT categorized as Major, Moderate, or Minor.

Show Minor Perform Warnings

An example of a Minor Perform Warning is a Backward PERFORM, which can slightly disrupt the following of PERFORM logic while analyzing COBOL logic OR possible identify another error in logic where a different PERFORM routine was intended. Turning this option on or off controls showing these Minor Perform Warnings.

Note - Perform warnings are also shown under the tab 'Perform Warnings'. However, when shown in that tab, the warnings are NOT categorized as Major, Moderate, or Minor.

Show COBOL Errors

CAT does lexical and syntactical analysis of a COBOL program similar to what a COBOL compiler does. However, the end result is that we do COBOL analysis rather than producing object code. We show these errors and warnings found within and at the end of the report produced.

These errors may also be seen by viewing the Alternate Compile Listing produced when running CAT by clicking on the ACL tools button. Turning this option on or off controls whether or not these errors are shown in the COBOL errors report.

Highlight Lines in Source Listing Detailed Description

Show Unprintable Characters (Light Blue)

There are some ASCII characters that are unprintable in a Rich Text box (format used) for the COBOL source listing. The sequence number, column number and length of the string of characters may be found by option within the COBOL errors reports. However, an option is available here to highlight those lines within the source listing for easy viewing. Clicking or selecting the option here will highlight these lines within the source listing.

Show Major Perform Warnings (Magenta)

CAT allows the option of showing Major Perform warnings in the 'Cobol Errors' tab report. Using the option listed here will highlight the source line within the Major Perform warning with the source listing window. See notes on Perform Warnings options under the 'Cobol Error' tab and/or see notes on Perform Warnings under the 'Perform Warnings' tab.

Show Moderate Perform Warnings (Light Yellow)

CAT allows the option of showing Moderate Perform warnings in the 'Cobol Errors' tab report. Using the option listed here will highlight the source line with the Moderate Perform warning within the source listing window. See notes on Perform Warnings options under the 'Cobol Error' tab and/or see notes on Perform Warnings under the 'Perform Warnings' tab.

Show Minor Perform Warnings (Light Magenta)

CAT allows the option of showing Minor Perform warnings in the 'Cobol Errors' tab report. Using the option listed here will highlight the source line with the Minor Perform warning within the source listing window. See notes on Perform Warnings options under the 'Cobol Error' tab and/or see notes on Perform Warnings under the 'Perform Warnings' tab.

Narrative Selection - 5th Tab

The Narrative Options contain two different sections including Data Division Options and Procedure Divisions Options as well as unrelated General options. The Narrative Window in CAT is there to provide full information on selected data names. To choose a data name for selection, simply click on a data-name (not paragraph name) anywhere in the source listing window. Data-name are highlighted (normally in **blue**) in the Data Division and also in the Procedure Division.

One of the aims of CAT ('Cool Auditing Tool' for COBOL) is to keep the programmer/technician or COBOL auditor within the Procedure Division as much as possible and one of the ways we do this is to allow a data-name to be quickly viewed with as much information on the data-name as the user wants to see.

Two types of information or documentation are provided in the Narrative Window for each selected data-name and these two types are Data Division information and Procedure Division references.

Individual options within the 'Narrative Selection' Tab allow the user to specify how much Data Division or how much or what type of format to use in showing Procedure Division references. Also for Procedure Division references, a special narrative condensed format is used to list each found Procedure Division COBOL statement. (For instance a # is used to indicate the data-name being referenced to save space in the Narrative window.)

Besides these Individual options under the 'Narrative Selection' Tab, there is a selection button (+/-) on top of the Narrative window that may be used at any time to go back and forth between just shown Procedure Division references OR also to show Data Division information as indicated in the options, shown underneath the 'Narrative Selection' Tab.

The screenshot shows a software interface with a tabbed menu at the top: 'Quick View', 'Perform Warnings', 'Cobol Errors', 'Options', 'Narrative Selection' (which is the active tab), and 'Debug'. Below the tabs, the 'Narrative Options Tab Panel' is displayed, divided into four sections:

- Data Narrative / Data Division Options:**
 - ☒ Include From-To Position within 01 Record
 - ☒ Include Data Div Section and COPY name
 - ☒ Include Message to Identify GROUP items
 - ☒ Show Level Num with COPY, Data Name, Pic, Usage and Value
 - ☒ Show when a Field REDIFINES another field
- Data Narrative / Procedure Divisions Options:**
 - ☒ Show Procedure Division Narrative
 - ☐ Show Narrative by Sorted P-D Verb (more condensed)
 - ☒ Show Narrative in Program Sequence (less condensed)
 - ☐ Show Indirect References for all non-level 01 fields
 - ☐ Temporarily Block Procedure Division Narrative
- General Options:**
 - ☒ Display Error Dialogs
- Special Options:**
 - ☐ Create CAT Stop Run

Narrative Options Tab Panel

Data Narrative / Data Division Options

Include From-To Positions and 01 Record Name

CAT calculates the correct From-To positions for each data-field relative to the 01 record that the data-field belongs to. For instance if Social Security is defined as 9 positions and appears first in a record, it's From-To positions in the record would be 1-9.

Checking this option indicates that both the From-To positions and the 01 record name will be shown in the Data Division part of the Narrative.

Include Data Division Section and any COPY name

When this option is checked the Data Division Section (File, Working-Storage, Linkage, etc.) that the field or data-name comes from will be included in the Data

Division Narrative for this data-name. Also, if the data-name is embedded in a COPY member, the name of the COPY member will be shown with the Data Division Section.

Include Message to identify GROUP items

When this option is checked, if the data-name is a GROUP item other than an 01 record, a line will appear in the Data Division Narrative to indicate this. A Group item is defined as a non-elementary item that contains one or more elementary fields.

Show Level Number, COPY Indicator, Data Name, Pic, Usage, and Value

When this option is checked, one OR more lines will be generated depending on the space needed to show the level number, COPY Indicator (a C after level number indicating this field is part of a COPY member), the Data-Name and PIC, USAGE, and VALUE clauses. These lines along with other lines created for the Data Division show in the Narrative Window OR NOT depending on the toggling of the (+/-) selection button above the Narrative window.

Show when a field REDEFINES another field

When this option is checked, a REDEFINE clause will be shown for fields where one is included.

Other Data Division Narrative not specified by option

Occasionally there is other Data Division generated for which there is no option specified, such as a Numeric field with decimal digits, the presence of an OCCURS clause, the presence of a 66 level entry. When CAT encounters this information it will include it for certain Data-Names and will appear when Data Division information is selected to appear in the Narrative window.

Data Narrative / Procedure Division Options

Show Procedure Division Narrative

The prime reason for having the Narrative window is to provide Procedure Division Narrative (or Procedure Division references for each data-name) so the user of CAT can stay focused on Procedure Division activity without needlessly chasing down what happens to each data-field on a continual basis. Turning this option off will temporarily block showing this Narrative and only show selected Data Division information and may be done if needed for some reason.

Show Narrative by Sorted P-D Verb (more condensed)

Selecting this option over the option to show Narrative in Program Sequence will often (though not always) condense the Procedure Division Narrative into less lines and will sort all references into Verb sequence and then into Program Sequence for each Verb. There will be only one occurrence of each Verb for any one data field and the sequence numbers for all occurrences will be listed within the COBOL Narrative for that Verb.

Show Narrative in Program Sequence (less condensed)

Selecting this option correlates all Procedure Division references in the order of their appearance in the Procedure Division. Whenever possible, verbs will be summarized together with sequence numbers showing each reference within the Narrative. When there is another Verb using this data field in the middle of a string of references (e.g. an IF in the middle of several MOVE's), extra lines of COBOL Narrative will be generated to accommodate the sequence of COBOL statements referencing this data field.

Show Indirect References for each field selected

Indirect references to a data field refer to ways that a data field can be modified, compared, or accessed without a direct reference to field. An example of this is Moving SPACES to an 01 record, where every field in the record is modified at the time of the MOVE. Indirect references also include REDEFINES. Also, Indirect references include any access to GROUP fields, or ELEMENTARY fields within a GROUP field, if the field selected is a GROUP field. Turning on this option adds an one or more extra lines, showing both the sequence number where the indirect reference occurs and ALSO whether the indirect reference is used to Change (modify), Use (access, but not modify), or Test (compare) this data field.

Temporarily Block Procedure Division Narrative

Turning on this option blocks all Procedure Division references (both Direct and Indirect) from being displayed in the Narrative window until this option is again turned off.

Alternate Compile Listing

Introduction to the Alternate Compile Listing Facility

The Alternate Compile Listing Facility of DCD III has one primary function, that is to provide an alternate listing from the one produced by the COBOL compiler with information provided in a way which will reduce substantially the time spent in maintaining a COBOL program.

The following DCD III reports are available through the Alternate Compile Listing Facility of CAT for replacing significant parts of the COBOL compiler listings:

1. Source Listing (Replaces compiler source listing)
2. Condensed VERB Report
3. CALL Statements
4. COPY Statements
5. Figurative Constants
6. Literals
7. Special Registers
8. Data Division Condensed Cross Reference (replaces compiler data name cross reference)
9. Procedure Division Condensed Cross Reference (Replaces compiler procedure name cross reference)
10. Unused Paragraphs and Sections
11. Unused 01 Records
12. Unused Data Names
13. Excess Indirect References

Some reports produced by the COBOL compiler (e.g. DMAP and PMAP reports) have no corresponding reports produced by the Alternate Compile Listing Facility of DCD III. Whether

using commercial packages such as COBOL/Aid, the CA-Optimizer or just running the compiler stand alone, features are available within DCD III on the mainframe version to produce one complete source listing for the COBOL programmer to work from.

The graphical interface CAT uses to view the Alternate Compile Listing includes tools to help navigate the source listing and reports with greater ease. These tools are discussed in detail after the reports. Samples of the reports produced by the Alternate Compile Listing Facility along with a description of each are provided in this manual. See 'Overview of Alternate Compile Listing Facility Reports'.

Overview of Alternate Compile Listing (ACL) Facility Reports

1. Source Listing

The Alternate Compile Listing Facility Source Listing is the main report. While other DCD III and compiler cross reference reports center around the Source Listing Report, most maintenance of the COBOL program in the Alternate Compile Listing or in CAT is done without consulting cross-reference reports.

Besides holding a listing of the COBOL program, the DCD III Source Listing Report also contains uniquely formatted cross-reference information. The presentation of this information is done in one format for the Data Division and is done in another format for the Procedure Division.

DATA DIVISION

In the Data Division part of the listing, formatting is done as follows:

1. A Procedure Division look-alike statement is generated.
2. The data name from the left hand side of the listing is represented by an # on the right hand side of the listing.
3. The compiler sequence number of the Procedure Division Statement(s) is put in parentheses.

For example, the statement:

```
05 WS-NEXT-COLUMN PIC S9 (4) COMP VALUE ZERO.      > Add 1 to # (1019)
```

indicates that a Procedure Division statement ADD 1 TO WS-NEXT-COLUMN will be found at compiler line number 1019 within the program.

If another data name is involved in the associated Procedure Division statement, then the compiler (or ACL) sequence number of that data name is given following an @ character.

For example:

```
05 WS-NEXT-COLUMN PIC S9(4) COMP VALUE ZERO.      > If # = WS-MAX-COLUMN @321 (1005)
```

If the same Procedure Division statement is used more than once, then multiple sequence numbers are provided. For example:

```
05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO.      > Move WS-MAX-PD-DIGITS @322 to # (939,962,  
1017)
```

In some instances, only a part of the Procedure Division statement is provided. For example:

```
05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO.      > Perform-Until # = ZERO (951,974,1031)
```

In most instances, however, the entire COBOL statement from the Procedure Division is shown. For example:

```
05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO.      > Subtract 1 from # (1040)
```

In many cases, where the Procedure Division statements are similar, but relate to different operands, the narrative will not reprint the left most part of the Procedure Division statement that does not change between the similar statements. For example:

```
05 WS-MAX-PD-DIGITS PIC S9(3) COMP-3 VALUE+5.      > If # =3 (942,965,1022,1050), 4 (945,968,  
1025,1064)  
Move 03 to # (497), 04 to # (501)
```

When no narrative appears along side a Data Division name, this indicates that the data-name is unreferenced.

When narrative is present, Indirect References will also be shown for procedure division activity that is present for redefined fields and other group fields and any field that has overlapping field positions. See the bolded line on next page.

* * *

```

SQ-NBR 1----6 PROGRAM-ID DCDLOGIC 01/01/01 14:56 PAGE 09 73----80 # REFERS TO DATA-NAME AT THIS SQ-NBR
320 014000 05 WS-NEXT-COLUMN PIC S9(4) COMP VALUE ZERO. > Add 1 to # (1019)
                                     If # > WS-MAX-COLUMN @321 (1005)
                                     Move 02 to # (1008)
                                     Set # to W34-INDEX @276 (1041)
                                     Set W34-INDEX @276 TO # (1020)
321 014100 05 WS-MAX-COLUMN PIC S9(4) COMP VALUE +34 > Indirectly Changed @ 318
                                     If WS-NEXT-COLUMN @320 > # (1005)
                                     If WS-MAX-PD-DIGITS @322 > # (1007)
                                     Move 34 to # (893)
322 014200 05 WS-MAX-PD-DIGITS PIC S9(3) COMP-3 VALUE +5. > Indirectly Changed @ 318
                                     If # = 3 (942,965,1022,1050), 4 (945,968,
                                     1025,1064)
                                     If # > WS-MAX COLUMNS @321 (1007)
                                     Move # to WS-NUMBER-TIMES @324 (939, 962,
                                     1017)
323 014300 05 WS-NARR-CODE-INC PIC S9(3) COMP-3 VALUE ZERO. > Indirectly Changed @ 318
                                     Compute S-WORK-NARRATIVE-CODE @299 = #
                                     (682, 694, 705)
                                     Move 03 to # (674), 06 to # (678), ZERO
                                     to # (670)
324 014400 05 WS-NUMBER-TIMES PIC S9(3) COMP-3 VALUE ZERO. > Indirectly Changed @ 318
                                     Move WS-MAX-PD-DIGITS @322 to # (939,
                                     962, 1017)
                                     Perform-Until # = ZERO (951, 974, 1031)
                                     Subtract 1 from # (1040)
325 014500 05 WS-RECORDS-TO-SORT PIC S9(9) COMP VALUE ZERO. Indirectly Changed @ 318

```

Exhibit 7

Source Listing Report - Data Division

Source Listing Report - Procedure Division

PROCEDURE DIVISION

The cross-reference information for the Procedure Division accommodates two types of information.

a. SQ-NBR'S

The first type references back to the Data Division (Procedure Division statements such as MOVE or ADD referencing a data name which resides in the Data Division). For these, ACL provides a sequence number referencing the line where the data name resides in the Data Division. For example:

```

1017 MOVE WS-MAX-PD-DIGITS TO WS-NUMBER-TIMES. 322 324

```

The first data name, WS-MAX-PD-DIGITS, is documented by the first number on the right, 322, which points to the Data Division sequence number where WS-MAX-PD-DIGITS resides. The second number, 324, references the Data Division sequence number where WS-NUMBER-TIMES resides.

b. LOGIC FLOW

The second type, references transfer of control statements such as GO TO or PERFORM, with references to or from another area of the program or statements which reference an external program. ACL handles these by providing appropriate narrative on the right and (if not external to the program) by providing a sequence number of where the transfer of control is going to or coming from. An example of a GO TO is provided here:

1028	GO TO B999-EXIT.	GO TO 1044
1044	B999-EXIT.	COMES FROM GO TO 1028

If the transfer of control statement is conditional on the action of an IF, AT END, or other conditional statement, then the four letters COND will precede the generated narrative. See the GO TO in Exhibit 8 on the next page.

PERFORMS are handled in the following manner:

1030	PERFORM B500-FLOAT-FL2-IN THRU B599-EXIT.	PERFORM 1036 THRU 1042
	.	
1036	B500-FLOAT-FL2-IN.	PERFORMED BY 1030
	.	
1042	B599-EXIT. EXIT.	RETURN TO PERFORM AT 1030

Statements that transfer control to or from another program are documented accordingly with statements such as:

PROCEDURE DIVISION.	BEGIN OF PROGRAM
CALL 'DCDPRINT'.	CALL TO EXTERNAL PROGRAM
GOBACK.	GOBACK

```

                                * * *
SQ-NBR 1-----6    PROGRAM-ID DCDLOGIC  01/01/2001 14:56  PAGE 23  73-----80  SQ-NBR'S      LOGIC FLOW

1017 033400  MOVE WS-MAX-PD-DIGITS TO WS-NUMBER-TIMES.          322 324
1018 033500
1019 033600  ADD      1          TO WS-NEXT-COLUMN.             320
1020 033700  SET W34-INDEX      TO WS-NEXT-COLUMN.             276 320
1021 033800
1022 033900  IF      WS-MAX-PD-DIGITS = 3                      322
1023 034000      SET FL2-INDEX TO 3                            292
1024 034100  ELSE
1025 034200      IF WS-MAX-PD-DIGITS = 4                      322
1026 034300      SET FL2-INDEX TO 2                            292
1027 034400      ELSE
1028 034500          GO TO B999-EXIT.                          COND GO TO 1044
1029 034600
1030 034700  PERFORM B500-FLOAT-FL2-IN THRU B599-EXIT          PERFORM 1036 THRU 1042
1031 034800      UNTIL          WS-NUMBER-TIMES = ZERO.        324
1032 034900
1033 035000 B299-EXIT EXIT.                                    RETURN TO PERFORM AT 880
1034 035100
1035 035200
1036 035300 B500-FLOAT-FL2-IN.                                  PERFORMED BY 950 973 1030
1037 035400  MOVE FL2-ALPHA-POS (FL2-INDEX) TO W34-POS (W34-INDEX). 291 292 275 276
1038 035500  SET FL2-INDEX  UP BY 1.                             292
1039 035600  SET W34-INDEX  UP BY 1.                             276
1040 035700  SUBTRACT 1      FROM WS-NUMBER-TIMES.             324
1041 035800  SET WS-NEXT-COLUMN TO W34-INDEX.                   320 276
1042 035900 B599-EXIT EXIT.                                    RETURN TO PERFORM AT 973 1030
1043 036000
1044 036100 B999-EXIT.                                          COMES FROM GO TO 1028
1045 036200  EXIT.                                             END OF OUTPUT PROCEDURE FROM 511

```

Exhibit 8

Source Listing Report - Procedure Division

2. Condensed VERB Report

The Condensed VERB Report shows all references for each VERB in the Procedure Division. Some verbs such as PERFORM are broken up into two categories (e.g. PERFORM & PERFORM THRU)

The Verb sequence numbers are listed in a string of numbers at the right of the VERB.

The following sequence is used for this report:

1. VERB
2. Sequence Number

See Exhibit 9 below for an example of the Condensed VERB Report.

VERB REPORT FOR PROG103

ACCEPT	1203	1569	2045						
ADD	982	995	1256	1290	1337	1498	1499	1503	1602
CLOSE	1572								
COMPUTE	1545	1547	1549	1551	1623	1625	1627	1629	1660
DISPLAY	2041	2042	2043	2163					1783
.									
.									
.									
STRING	1802								
SUBTRACT	961	984							
WRITE	1337	1924	1927	2011	2013				

Exhibit 9

Condensed VERB Report

3. CALL Statements

The CALL Statement Report shows all ENTRYs into a program and CALLs out of the program in a concise report provided for quick and easy analysis of CALL activity within a program.

Each ENTRY or CALL is listed on one line. The number of parameters associated with that ENTRY or CALL is also listed on the same line along with a sequence number for finding the ENTRY or CALL within the program.

The following sequence is used for this report:

1. CALL name (or ENTRY name)
2. Sequence number in the Procedure Division of where the CALL or ENTRY is found.

4. COPY Statements

The COPY Statements Report gives a listing of all COPY members used within the program and for CICS, DL1 & DB2 programs give a listing of all INCLUDE members.

Each COPY statement is listed on one line. Additional information associated with the COPY statement such as SUPPRESS, the REPLACING clause and the clause OF

LIBRARY-NAME have their presence indicated as shown in Exhibit 9. The division where the COPY statement resides is also shown, as is the sequence number of where the COPY statement may be found.

The following sequences are used for this report:

1. COPY text-name
2. Sequence number of where the COPY statement resides

See Exhibit 10 below for an example of CALL & COPY statement formatting.

CALL STATEMENTS

478	PROGRAM	ENTRY	DCDLOGIC	USING 1	PARAMETER
482	CALL		DCDPRINT	USING 2	PARAMETERS
653	CALL		DCDPRINT	USING 2	PARAMETERS
702	CALL		DCDPRINT		
552	CALL		DCDSTAMP		

COPY STATEMENTS

DATA DIVISION	344	COPY	DCDLINK	SUPPRESS		OF LIBRNAME
DATA DIVISION	105	COPY	PARATYP2		REPLACING	
DATA DIVISION	204	COPY	PDNARTBL			

INCLUDE STATEMENTS

DATA DIVISION	356	INCLUDE	PDINCLUD
DATA DIVISION	219	INCLUDE	SALESTB3

Exhibit 10 CALL and COPY Statements Reports

5. Figurative Constants

The Figurative Constants Report lists occurrences of each Figurative Constant by sequence number of where they occur in the program.

Figurative Constants include the following: ZERO(S) (ES), SPACE(S), HIGH-VALUE(S), LOW-VALUE(S), QUOTE(S) and ALL literals.

For purposes of listing the Figurative Constants, the plural and single form of the word are treated and listed as one. For example, ZERO, ZEROS and ZEROES are listed as ZERO(S) on the report.

ALL literals are listed in the Literals Report, not in the Figurative Constants Report. The Literals Report directly follows the Figurative Constants Report. Any occurrences of the ALL literals are listed first within the Literals Report. See Exhibit 10.

The following sequence is used for this report:

1. Figurative Constants in alphabetic order
2. Sequence number of where the Figurative Constant is used.

See Exhibit 11 for an example of formatting of the Figurative Constants Report.

```

* * *
FIGURATIVE CONSTANTS
LOW-VALUE(S) 095 097 486 493
SPACE(S)      098 099 328 330 322 334 336 394 487 494 568 633 645 662
              711 722 759 771 783 800 833 886 1018
ZERO(S)       101 171 172 173 316 317 318 319 320 323 324 325 630 638
              667 671 681 704 708 715 719 764 776 780 797 830 943 962

```

Exhibit 11

Figurative Constants Report

6. Literals

The Literals Report lists all occurrences of ALL literals, nonnumeric literals within quotes and numeric literals by sequence number of where they occur in the program.

ALL literals items are listed first, putting them closer to the Figurative Constants Report just above or ahead of this report.

Thirty characters are provided for listing the non-numeric literals including the quotes that surround them. If the literal is not over 28 characters, then the entire non-numeric literal is shown. If the literal is over 28 characters, only the first 26 characters are shown, followed by two periods and the ending quote. See Exhibit 11 for two examples of a literal over 28 characters long.

The following sequence is used for this report:

1. ALL literals are placed ahead of other literals
2. Literal value (for non-numeric literals, the surrounding quotes also figure in the sorting sequence)
3. Sequence number of where the literal is used

See Exhibit 12 for an example of the Literals Report.

LITERALS			*	*	*								
ALL '9'	849	850	858	864	1007								
+34	321												
+5	322												
'PF2'	486												
'PF3'	493												
'000 UNUSED ..'	254	256											
'0726 ALTER NNNN TO PROCED..'	1082												
'1'	369	397											
'2'	370												
'3'	371												
'A'	398												
'COND'	1098												
'DCDLOGIC END WS DCDLOGIC'	340												
'E'	457												
'1'	424												
'N'	383	502	871	888	892								
'0'	460												
'Y'	144	147	329	331	333	335	337	353	356	358	361		
	375	378	382	386	389	393	404	407	410	419	426		
	442	445	472	537	619	742	851	859	893	906	932		
01	178	627	896										
1	959	982	1030	1039	1049	1050	1051						
2	957	980	1037										
	953	954	976	977	1033	1034	1061						

Exhibit 12

7. Special Registers

Special Registers refer to compiler generated areas that may be used in conjunction with specific COBOL features.

1. CURRENT-DATE
2. DATE
3. DEBUG-ITEM
4. DEBUG-SUB-1
5. LINAGE-COUNTER
6. RETURN-CODE
7. SORT-RETURN
8. TALLY
9. TIME

Only those Special Registers found within the program will be listed in this report. If no Special Registers are used, the report will not appear.

The following sequence is used for this report:

1. Special Registers
2. Sequence number of where the Special Register is used.

See Exhibit 13 for an example of the Special Registers Report.

```

                                * * *
SPECIAL REGISTERS
CURRENT-DATE 744
RETURN-CODE  604 843 849 1283
TIME-OF-DAY  746
```

Exhibit 13

Special Registers Report

8. Data Division Condensed Cross Reference

The Data Division Condensed Cross Reference Report provides an alphabetic listing of the data names within the Data Division.

What is unique about this report is that no references are given to the Procedure Division statements that reference the data name. The only sequence number given points to the field's location within the Data Division. The cross reference given at that location within the Source Listing Report gives, in COBOL, everything that happens to this field.

Besides listing the data name and the sequence number of its location, four characters are reserved alongside the listing to provide the following information about the data field:

1. The level number (**01-49, etc.**) associated with this field
2. The letter **G** for those fields which are GROUP items
3. The letter's **INDX** for index items
4. The letter's **FILE** for file names

It is recommended that this report be used in place of the compiler cross reference when using the DCD III Source Listing Report for the following reasons:

1. Source Listing Report when using the ACL Source Listing Report, the Cross Reference Report is used much less.
2. The narrative alongside each field in the Data Division provides a complete cross reference for it.
3. The Condensed Cross Reference saves approximately 70% of the paper generated by a compiler cross reference.

The following sequence is used for this report:

1. Data name
2. Sequence number of the data name

See Exhibit 14 for an example of this report.

01/01/2001 15:26		DATA DIVISION CONDENSED CROSS REFERENCE FOR DCDLOGIC				PAGE 27		
05	0316	AL-ALTER-CMPLR-NBR	:	88	0370 LINK-PRINT-JCL-DSNAME	:	88	0120 PAR-2-LAST-TOKEN-IN-PROG
05	0317	AL-ALTER-COLUMN-NBR	:	88	0375 LINK-PRINT-LAYOUTS	:	88	0147 PAR-2-MULTIPLE-OPER...
05	0318	AL-GO-TO-CMPLR-NBR	:	20	0374 LINK-PRINT-LAYOUTS-SW	:	05	0146 PAR-2-MULTIPLE-OPERAND
05	0319	AL-PARA-CMPLR-NBR	:	88	0378 LINK-PRINT-RECORDS05	:	05	0150 PAR-2-NAME-COLUMN-NU
G 01	0092	CAFE-DATA-ENVIRON-NAME	:	20	0377 LINK-PRINT-RECORDS-SW	:	05	0149 PAR-2-NAME-EXP-LINE-NUMBR
01	0089	DCDLOGIC-BEGINS-HERE	:	88	0382 LINK-PRINT-SOURCE-LISTING	:	88	0129 PAR-2-OUTPUT-PROCEDURE
01	0339	DCDLOGIC-ENDS-HERE	:	20	0381 LINK-PRINT-SOURCE-LISTING-SW	:	88	0130 PAR-2-OUTPUT-PROCEDURE-TH
10	0282	FL-ALPHA-POS	:	88	0386 LINK-PRINT-SYS-REC-ANAL	:	05	0112 PAR-2-PARA-OR-SECT
INDX	0283	FL-INDEX	:	20	0385 LINK-PRINT-SYS-REC-ANAL-SW	:	88	0117 PARA-2-PARAGRAPH-NAME
G 05	0281	FL-NUMBER-ALPHA	:	88	0393 LINK-PRINT-UNREF	:	88	0125 PAR-2-PERFORM
05	0280	FL-NUMBER-NUMERIC	:	88	0389 LINK-PRINT-VERB-ANALYSIS	:	88	0126 PAR-2-PERFORM-THRU
G 01	0279	FL2-NUMBER-RECORD	:	20	0388 LINK-PRINT-VERB-ANALYSIS-SW	:	88	0119 PAR-2-PRIOR-TOKEN-OF-PARA-S
10	0291	FL2-ALPHA-POS	:	05	0419 LINK-PROGRAM-NAME	:	88	0138 PAR-2-PROGRAM-ENTRY
INDX	0292	FL2-INDEX	:	05	0458 LINK-QUOTE-VALUE	:	G 01	0106 PAR-2-RANGES-FOR-PARAS
G 05	0290	FL2-NUMBER-ALPHA	:	G 01	0345 LINK-RECORD	:	05	0114 PAR-2-RELEVANT-SECTION
05	0287	FL2-NUMBER-NUMERIC	:	G 05	0438 LINK-RELEASE-DATE	:	88	0118 PAR-2-SECTION-NAME
G 01	0286	FL2-NUMBER-RECORD	:	10	0439 LINK-RELEASE-MONTH	:	05	0152 PAR-2-SOURCE-BEGIN-NBR
05	0288	FL2-NUMBER-ZZ-999	:	10	0441 LINK-RELEASE-YEAR	:	05	0155 PAR-2-SOURCE-END-NBR
10	0348	LNK-100	:	G 15	0350 LINK-REPORT-OPTIONS-1-25'	:	05	0108 PAR-2-SOURCE-LINE-NUMBER
G 10	0349	LNK-100-R1	:	05	0423 LINK-RETURN-CODE	:	88	0134 PAR-2-STOP-RUN
G 10	0404	LNK-100-R2	:	05	0427 LINK-SORT-REGION	:	05	0110 PAR-2-TOKEN-COLUMN-NBR
05	0445	LNK-ADDR-OF-DCDPRTLN	:	88	0398 LINK-SYS-REC-REC-ANAL-ALL	:	05	0116 PAR-2-VERB-CODE
88	0417	LNK-BASIS	:	88	0400 LINK-SYS-REC-ANAL-DIR-INDIR	:	05	0162 PAR-3-EXPANDED-LINE-NUMBE
05	0416	LNK-BASIS-SW	:	88	0397 LINK-SYS-REC-ANAL-ONLY-01	:	05	0166 PAR-3-FILLER
88	0414	LNK-COMPILE-MODE	:	88	0401 LINK-SYS-REC-ANAL-ONLY-DIR	:	05	0164 PAR-3-NARRATIVE
05	0413	LNK-COMPILE-MODE-SW	:	20	0396 LINK-SYS-REC-ANAL-SW	:	G 01	0160 PAR-3-NARRATIVE-FOR-SR-LST
88	0456	LNK-COPY-REPLACE-FOUND	:	88	0430 LINK-SYSTEM-EXPIRED	:	FILE	0056 PARA-FILE-2
05	0455	LNK-COPY-REPLACE-FOUND-SW	:	05	0429 LINK-SYSTEM-EXPIRED-SW	:	FILE	0065 PARA-FILE-3
88	0383	LNK-DO-NOT-PRINT-SRC-LIST	:	05	0443 LINK-TIME-OF-DAY	:	01	0098 PARA2-NAME
88	0394	LNK-DO-NOT-PRINT-UNREF	:	88	0433 LINK-TRIAL-TAPE-EXPIRED	:	01	0099 PARA3-NAME

Exhibit 14

Data Division Condensed Cross Reference

9. Procedure Division Condensed Cross Reference

The Procedure Division Condensed Cross Reference Report provides an alphabetic listing of the paragraph/section names within the Procedure Division. If sections are also used, a separate report shows just SECTIONS and is shown at the end of the Paragraph/Section report.

This report is similar in nature to the Data Division Condensed Cross Reference Report in that no references are given to Procedure Division statements which reference the paragraph or section. The only sequence number given points to its location within the Procedure Division. If the paragraph or section is referenced by other Procedure Division statements such as GO TO or PERFORM, narrative will appear alongside the line in the Source Listing Report where the paragraph or section resides. See Exhibit 8 examples of the narrative that appears alongside the Procedure Division lines that contain a paragraph or section (i.e., lines 1033, 1036, 1042, and 1044).

The format of this report, like the Data Division report, has four characters to provide additional information as follows:

1. For paragraph names, these four characters are blank
2. For section names, these four characters contain the letters - **SECT**

Like the previous report, it is recommended that this report be used over the Compiler Procedure Name Report for the same reasons given there, among them a 70% savings in paper.

The following sequence is used for this report:

1. Paragraph/Section name
2. Sequence number of the procedure name

See Exhibit 15 for an example of this report.

0470 003-BUILD-ASSIGN-NAMES	:	0640 A300-GO-TO-DEPENDING	:	0930 B140-TEST-FOR-NUM-1-FLOAT
0494 010-REBUILD-PDNARTBL-COPY	:	0666 A399-EXIT	:	0953 B150-TEST-FOR-NUM-2-FLOAT
0504 020-SORT-DATA-NAMES	:	0668 A400-PERFORM-TYPE	:	0976 B170-MOVE-IN-NARRATIVE
0514 090-GOBACK	:	0714 A499-EXIT	:	0983 B180-WRITE-NARR-RECORD
SECT 0517 A-BUILD-WORK-RECORDS	:	0717 A500-ALTER-PARA	:	0987 B1980-RETURN-NEXT-RECD
0518 A 010-OPEN-PARA-FILE-2	:	0775 A599-EXIT	:	0999 B199-EXIT
0522 A020-READ-FIRST-RECORD	:	0778 A600-CANCEL-PROGRAM	:	1003 B200-FLOAT-IN-NEXT-NUMB
0528 A030-BUILD-WORK-RECORDS	:	0792 A699-EXIT	:	1011 B210-SET-UP-FOR-FLOAT
0533 A040-CLOSE-PARA-FILE-2	:	0794 A700-ALL-OTHERS	:	1033 B299-EXIT
0537 A050-END-OF-INPUT-PROCEDURE	:	0825 A799-EXIT	:	1036 500-FLOAT-FL2-IN
0543 A100-BUILD-WORK-RECORDS	:	0827 A999-EXIT	:	1042 B599-EXIT
0544 A110-GO-TO	:	SECT 0828 B-BUILD-NARRATIVE-RECORDS	:	1044 B999-EXIT
0561 A120-GO-TO-DEPENDING-ON	:	0830 B010-OPEN-FILE-A	:	SECT 1047 C-REBUILD-PDNARTBL-COPY
0567 A130-PERFORM-TYPE	:	0834 B020-RETURN-FIRST-SORT-REC	:	1048 C010-TEST-MAX-OF-3
0578 A140-ALTER-PARA	:	0850 B030-PROCESS-ALL-RECORDS	:	1062 C020-TEST-MAX-OF-4
0584 A160-CANCEL-PROGRAM	:	0843 B030-RETURN-SECOND-SORT-REC	:	1076 C999-EXIT
0590 A170-ALL-OTHERS	:	0862 B040-CLOSE-FILE-3	:	SECT 1078 D-CHECK-FOR-INSERT-COND
0601 A190-READ-NEXT-RECORD	:	0866 B050-END-OUTPUT-PROCEDURE	:	1079 D010-CHECK-LENGTH
0610 A199-EXIT	:	0870 B100-PROCESS-ALL-RECS	:	1084 D020-INSERT-COND
0613 A200-GO-TO	:	0907 B120-CHECK-MATCH-TO-PREV-REC	:	1097 D999-EXIT
0637 A299-EXIT	:	0926 B130-SET-UP-EXP-LINE-NBR	:	

SECT 0517 A-BUILD-WORK-RECORDS	:	SECT 1047 C-REBUILD-PDNARTBL-COPY
SECT 0828 B-BUILD-NARRATIVE-RECORDS	:	SECT 1078 D-CHECK-FOR-INSERT-COND

Exhibit 15

Procedure Division Condensed Cross Reference

10. Unused 01 Records

01 Records that are completely unused are listed in this report along with unused 77 level entries.

Redefined 01 records are shown independently from the 01 record that it redefines.

Sometimes a beginning 01 record is not referenced and is used for setting up VALUES for a redefined 01 record that follows it. If the first 01 record is not referenced and the next redefined 01 record is referenced, the first 01 record is shown as not referenced and then a CAUTION line is printed showing that the next 01 record is used! Conversely, it is possible to have unused 01 records **follow** to add space to a previous table. These records should normally not be deleted.

If the first FD 01 record is not unused, including having no OPEN, READ, CLOSE, etc. to the associated FILE name with the FD, then the 01 record is shown as unused, otherwise it will be treated as used. A second, third or more FD 01 record for the same FD will be shown as unused if none of the fields in the respective record, including that 01, are unused.

01 records are shown as unused whether or not they reside in a COPY member.

11. Unused Data Names

Only unused 02-49 level entries are shown in this report. 01 record names, 77 level entries, and 88 level entries are not shown. FILLERs are not shown as unused. 02-49 level entries underneath a 66 level RENAMES clause are shown.

If the entire 01 record was shown above as unused, then individual 02-49 entries are not shown.

Also, 02-49 level entries within a COPY member are not shown as unused, as it is expected that most names in a COPY are not necessarily used in any one program. To clean up COPY book records, use the System Record Analysis report in the Tracing Analysis & Other Reports section that is available in mainframe DCD III software from Marble Computer, Inc.

When all of the fields within a GROUP field are unused, then the GROUP field will be shown as unused and the individual fields will not be listed. The only time the group field will not be shown is when a larger group field containing this group field is unused.

Caution must be used before removing a field that is shown as unused. The field may be named for documentation where a FILLER could be used and the field could be necessary to proper running of the program. To prevent against removing a field that should be left in place, Indirect References are shown after each field that is shown as unused. The line numbers of where the other fields are that have overlapping field positions to this field are listed. If no Indirect References are listed then the field may be removed. If Indirect References are shown, then they must be examined carefully before removing the listed field.

The 01 record associated with each field listed in this report is shown with its line number to assist in examining a field.

12. Unused Paragraphs and Sections

Un-referenced Paragraphs and Sections are shown for the entire program, with the exception of those within a DECLARATIVES SECTION. All paragraphs and sections that are present in any Declaratives code are accepted as used and will never be reported as being unused.

The first tag (beyond any Declaratives Section, if present), whether it is a SECTION or a PARAGRAPH name, is treated as an implied fall through and is never treated as un-referenced. If the first tag in the program is a SECTION then all paragraphs in that SECTION are treated as a fall through up until the first found STOP RUN, GOBACK, or EXIT, following a coded period within that section. If the first tag in the program is a paragraph name, then all paragraphs are treated as a fall through up until the first found STOP RUN, GOBACK, or EXIT following a coded period or the first found SECTION.

Option UPI (permanently defaulted to in CAT) is used to print this report, and all paragraphs or sections within a PERFORM range, whether it is a perform of a section or a range of paragraphs or sections, will not be listed as unused, but will be treated as being used in the range of the PERFORM. To show these paragraphs or sections, contact Marble Computer for assistance to undo the option UPI.

SECTIONS, if present and found to be not used, are listed in their own report ahead of PARAGRAPHS.

When a section is shown as being un-referenced, all paragraphs within this section will also be shown as un-referenced within the Paragraph report.

If a SECTION is otherwise un-referenced, but there is either an ENTRY verb into the section or there is a reference to a paragraph within the SECTION, then the SECTION will be listed as un-referenced and an additional line will immediately follow to show the presence of one or more ENTRY verbs and/or the presence of paragraphs being referenced.

If a PARAGRAPH is un-referenced, but there is one or more ENTRY verbs within the paragraph, then the paragraph will be shown as un-referenced and an additional line will immediately follow to show the presence of one or more ENTRY verbs being present.

Before removing any SECTIONS or PARAGRAPHS, it is up to the user to insure that a fall through does not exist or perhaps the routine's access is commented out and needs to be kept.

If a fall through exists:

If a fall through is intended, then it is recommended that this logic be well documented as to why. If it was not intended, there is the possibility of an existing PERFORM Error and the code should be

looked at closely. If the user is unsure whether or not a fall through may exist within the logic, one suggestion is to test for it by adding an additional un-referenced paragraph with a single DISPLAY just ahead of the paragraph or section that shows as being un-referenced. If the DISPLAY is invoked, during any test or production running of the program, then that is proof that a fall through exists and the user is cautioned to look more seriously at why this exists.

If a fall through does not exist, then the paragraph or section with all code may be removed!

13. Excess Indirect References

01/01/2001	15:26	INDIRECT REFERENCES	(See ACL Report for fields not shown here)	PAGE 31
Data-Name	SEQ-#	Indirect Access for this Field		
LINE-28	00025	Indirectly changed @ 40,60,80,102,103,104,106,107,108,109,111,117,121,125,129,13,133,134,137,138 Tested @ 105,111, Used @ 110,111,117,121,135,129,130,133,137,138		

Exhibit 16

Overflow report to Indirect References

Alternate Compile Listing Navigation Tools

Navigating CAT's Alternate Compile Listing is greatly facilitated by certain tools available in its interface. These tools are accessible via a toolbar at the top of the window.



Alternate Compile Listing Toolbar

The text box labeled “Goto Seq #” allows the user to enter a sequence number and click on the arrow button to go to the indicated line, which will be highlighted. The drop box labeled “Goto Following” contains a list of different parts of the Alternate Compile Listing; clicking on one of these will instantly go to it in the listing. The “Search” text box allows the user to search for any text in the listing by entering it in the text box and clicking the arrow button.

The blue backward button, on the left side of the toolbar, takes the user back to the line that was previously jumped to. The blue forward button to its right reverses the effect of the backward button. If the user clicks the backward button twice and the forward button once, the ACL display will jump backward twice and forward once. If the user clicks forward again, the ACL display will return to its original state, as if the backward/forward buttons hadn't been used. The red “X” between them erases the list of line jumps. Immediately after clicking the red “X”, the user will not be able to jump forward or backward; a list of jumps must first be built by jumping to lines with the Goto Seq # option, the Goto Following option or the Search option.

```

Alternate Compile Listing
Goto Seq # 47 Goto Following Search
USED PARM FIELD 06/21/2012 10:37 PAGE 01
DNF,FTB,LNC=047,

PARM OPTIONS IN EFFECT FOR THIS PROGRAM
ASKIPS (FOR SKIP1, 2, 3) *****
BF05 (MATCHES TO BUFNO) *
NOBRT -DO NOT BUILD BUS RULES * DCD III Release 3.8-1060 *
CALL -CALL REPORT *
DYCALLS (DYNAMIC CALL INFO) * -- A product of -- *
CICS (ALLOWED) *
COPY -COPY REPORT * MARBLE Computer, Inc. *
DB2 (ALLOWED) * P.O. Box 920692 *
DDC -DATA DIV SXREF * El Paso, Texas 79902 *
DLI (ALLOWED) *
ERRORS Errors are printed *
DNF give Warnings for missing names * Phone: 1-800-252-1400 *
FGCONSTS -FIGURATIVE CONSTS * 1-915-845-0963 *
FTB * SEE NOTE BELOW * Fax: 1-915-845-7918 *
NOIDD NO DATA DIV LITERALS *
INDEPENDENT (MODE) * Marble programs may not be *

```

Alternate Compile Listing before going to Sequence Number

Goto Seq #

```

Alternate Compile Listing
Goto Seq # 47 Goto Following Search
1SQ-NBR 1----6 PROGRAM-ID DEMOPGM 06/21/2012 10:37 R=1060 PAGE 04 73----80 # REF:
37 #REND
38 003300 DATA DIVISION. DEMOPROG
39 003400 FILE SECTION. DEMOPROG
40 003500 FD INV-MAST-FILE DEMOPROG> Close # (2.
Open Input
41 003600 RECORD CONTAINS 97 CHARACTERS DEMOPROG
42 003700 BLOCK CONTAINS 0 RECORDS DEMOPROG
43 003800 LABEL RECORDS ARE STANDARD.
44 003900 01 INV-1-RECORD-IN. 12/92 > Read # (20.
Indirectly
61,64 Us,
55,56,59
45 004000 05 INV-1-PART-NBR PIC S9(8). 1-8 > Move # to :
320,398)
Indirectly
46 004100 05 INV-1-PART-NAME PIC X(20). 9-28 > Move # to :
399)
Indirectly
47 004200 05 INV-1-ORIG-DATE. 12/92 > Move # to

```

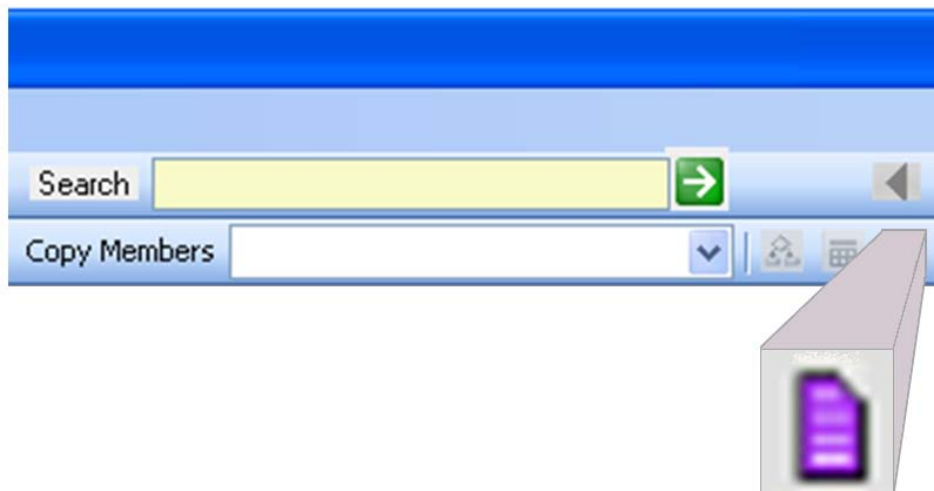
Alternate Compile Listing after going to Sequence Number

CAT Editor

CAT includes a source code editor that allows the user to modify COBOL code at the same time that other windows provide supporting analysis. Since the Syntax Editor is built as a separate window from the main interface, the user can work on one part of a program with the editor while viewing another part of the code with CAT analysis, hierarchy charts and more.

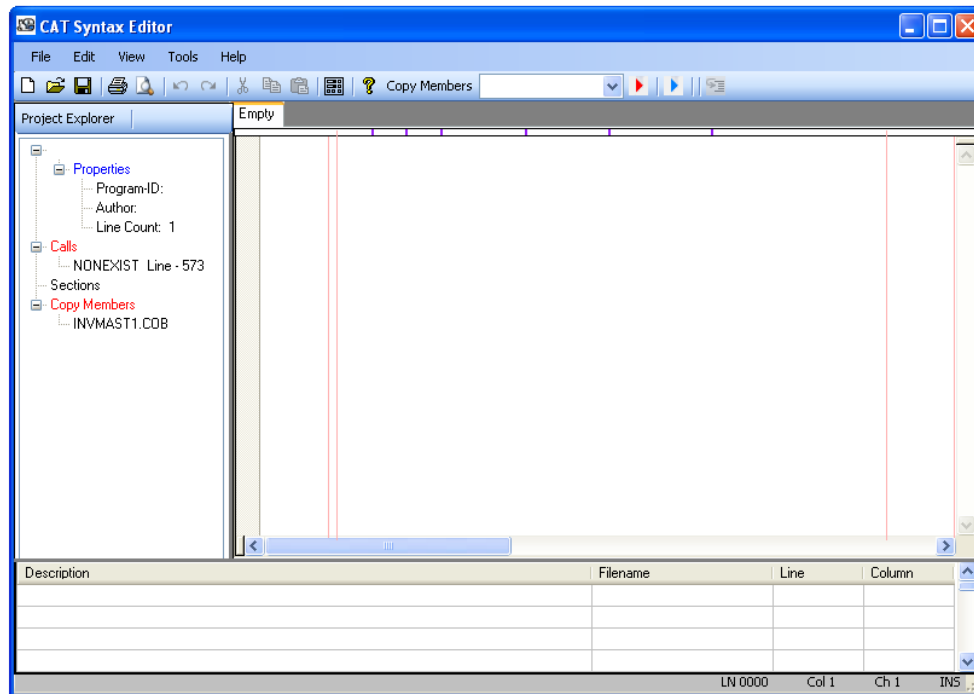
Launching the COBOL Editor

The Syntax Editor is launched from CAT, from a purple button to the right of the Hierarchy Chart button. See example below.



The editor loads with the program that is already loaded in CAT; if there is no program loaded, it loads a skeletal program or an empty page.

The source code window has visual delimiters to help the user keep track of columns 1-6, column 7 and columns 73-80. Margin B is identified by a tab stop.



CAT Syntax Editor window

The editor consists of seven main components –

- The menu bar
- The toolbar
- The tabs
- The project explorer window
- The source code window
- The errors window
- The status bar

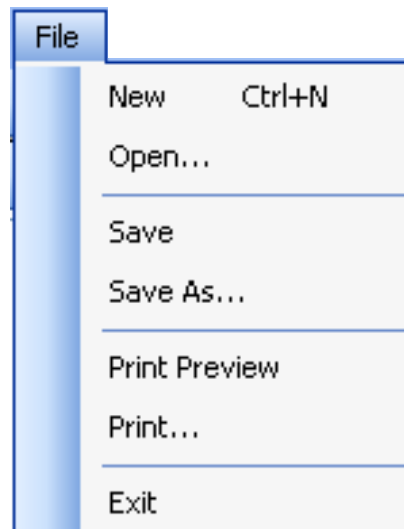
Menu bar

The Syntax Editor's functions are available via its menu bar, at the top of the editor window.



Syntax Editor menu bar

The options available in the menu bar are shown below.



File Menu

New

The New command allows opening a new program or Copy Member. Optionally a new tab may be opened, allowing several programs and/or Copy Members to exist at the same time in different tabs. The Editor can handle up to eight tabs at a time. There is another 'New' button on the toolbar, as well as a keyboard shortcut – *Ctrl + N*.

The New command can open a blank tab or a skeletal COBOL program, depending on the option selected in the New File Setup dialog box.

Open...

The Open command allows opening an existing program or Copy Member. Optionally a new tab may be opened allowing several programs or Copy Members to exist at the same time in different tabs. There is another 'Open' button on the toolbar.

Save

To save a file that already exists on disk, use the Save command or the toolbar button.

Save As...

To save a file for the first time, or to save it with a different name or in a different directory, use the Save As... command.

Print Preview

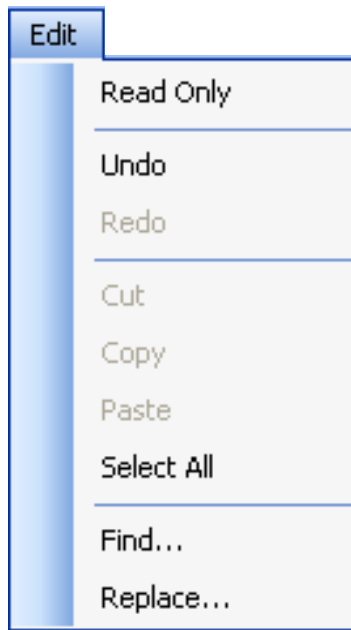
To see how the printed pages will look before printing, use the Print Preview command. There is also a PrintPreview button on the toolbar.

Print...

To print the currently selected tab's contents, use the Print command or the toolbar button.

Exit

To close the Editor at any time, use the  button or the 'Exit' option under 'File'.



Edit Menu

Read Only

The Read Only option prevents the user from modifying code within the editor.

Undo

Reverse a mistaken command in the Editor with the Undo command, the Undo button on the toolbar or the *Ctrl* + *Z* keyboard shortcut.

Redo

To reverse a mistaken Undo command, use the Redo command or the Redo toolbar button.

Cut

Cut text out from the Editor onto the Clipboard by selecting the text, then using either the Cut command, the Cut button on the toolbar or the keyboard shortcut *Ctrl* + *X*.

Copy

Copy text from the Editor onto the Clipboard by selecting the text, then using either the Copy command, the Copy button on the toolbar or the keyboard shortcut *Ctrl* + *C*.

Paste

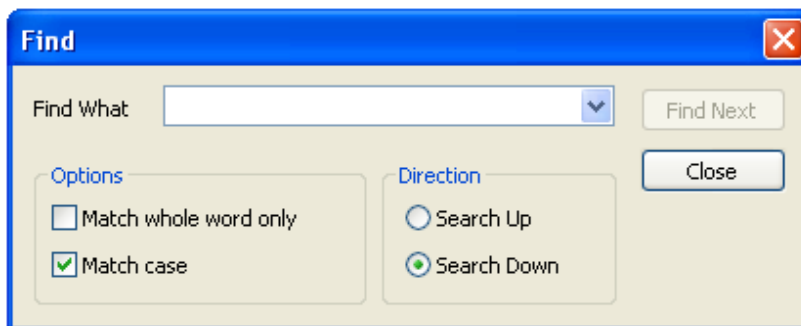
Paste text from the Clipboard into the Editor via either the Paste command, the Paste button on the toolbar or the keyboard shortcut *Ctrl* + *V*.

Select All

The Select All command selects (highlights) the entire program. This allows the whole program to be easily copied or cut.

Find...

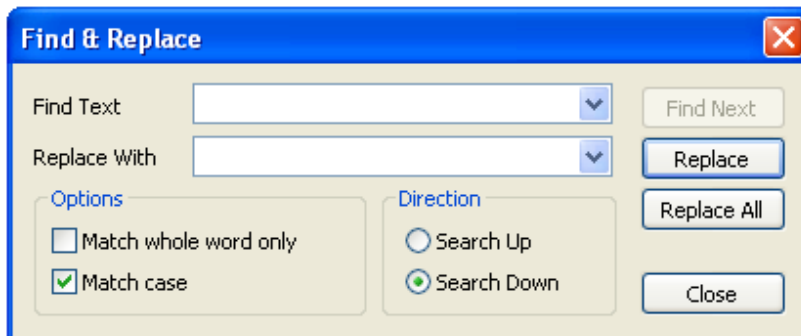
To search for text in the currently selected tab, click on Find....



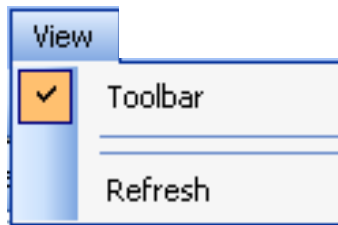
Find

Replace...

To search for text and replace it, click on Replace....



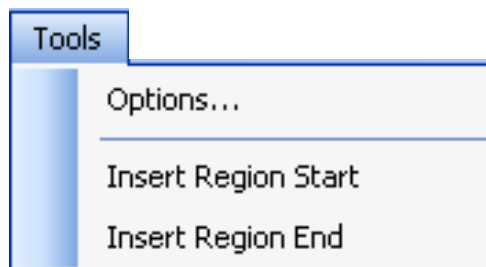
Replace



View Menu

Toolbar

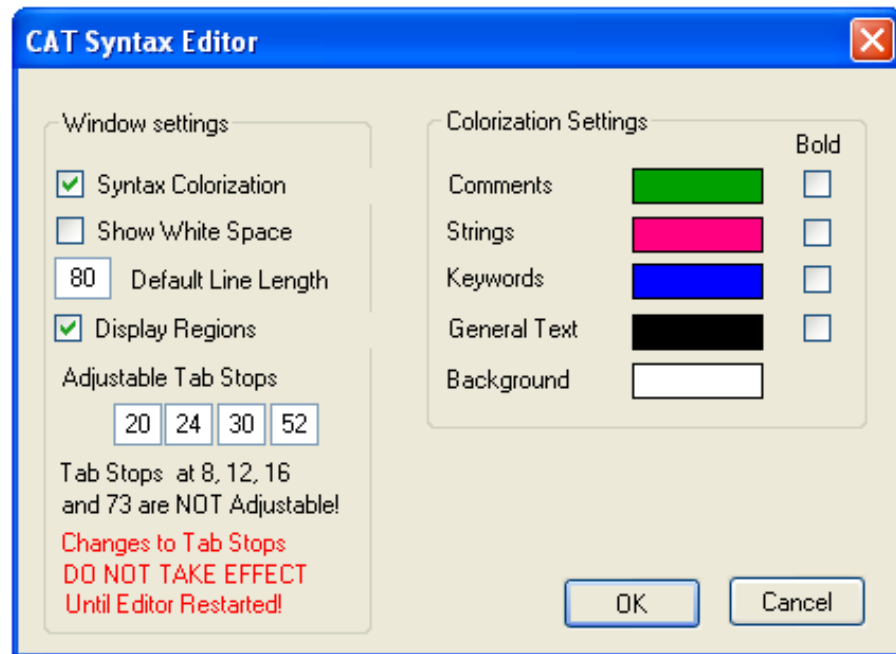
The Toolbar option displays and hides the toolbar.



Tools Menu

Options

To adjust the Editor's settings, click on Options.



Syntax Editor options dialog

Syntax Colorization check-box

This box should be checked to allow user colors within the editor.

Show White Space check-box

This box should be checked to mark every white space in the editor with a raised dot.

Display Region check-box

This box should be checked to allow collapsing regions.

Adjustable Tab Stops

Tab stops can be adjusted here. The Editor always has tab stops at columns 8, 12, 16 and 73; it also has four adjustable tab stops.

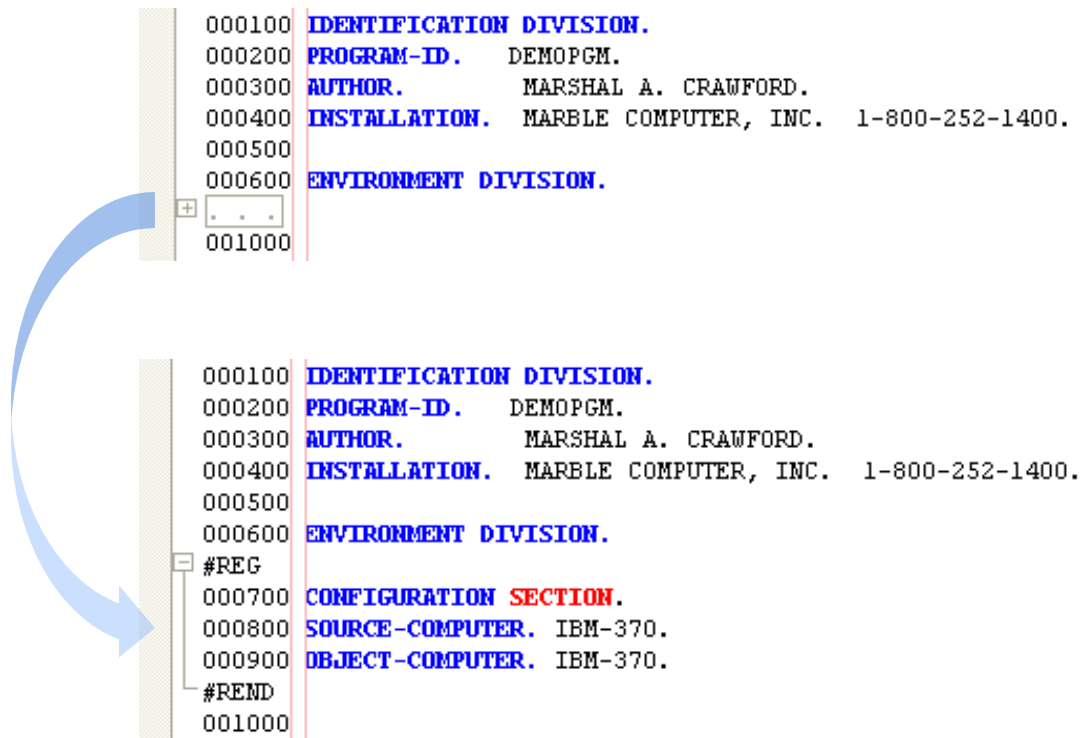
Note that the tab creates multiple spaces, not a tab character.

Colorization Settings

Four types of Cobol syntax can be differentiated by color in the editor – comments, text literals, reserved words and all other Cobol code. The different colors used for each part of syntax are selected here, as well as the editor's background color.

Regions

The Syntax Editor handles collapsible regions of code. These regions can be defined with the 'Insert Region Begin' and 'Insert Region End' options under 'Tools', or by typing in '#REG' in columns 1 – 4 of the line where the region should begin and '#REND' in columns 1 – 5 of the line where the region should end. (To expand a region, click on the plus sign (+); to collapse the region again, click on the minus sign (–) that appears in front of #REG.)



Regions

Toolbar

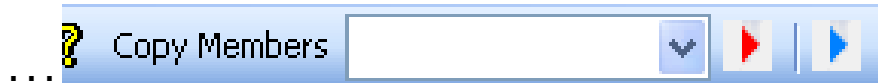
Shortcuts to many of the Editor's functions are available through the toolbar, found just under the menu bar.



Syntax Editor toolbar part 1

The first twelve buttons are, from left to right,

- | | | | |
|---------|-----------------|---------|-----------|
| 1. New | 4. Print | 7. Redo | 10. Paste |
| 2. Open | 5. PrintPreview | 8. Cut | 11. Find |
| 3. Save | 6. Undo | 9. Copy | 12. About |



Syntax Editor toolbar part 2

Copy Members box

The Copy Members drop-down box contains the names of the COPY members used in the COBOL program. Clicking on one of these COPY member names will give the option of opening the COPY member in a new tab. The COPY members are detected when the program is run in CAT. Note that the Syntax Editor can work with up to eight tabs at a time.

Red arrow button

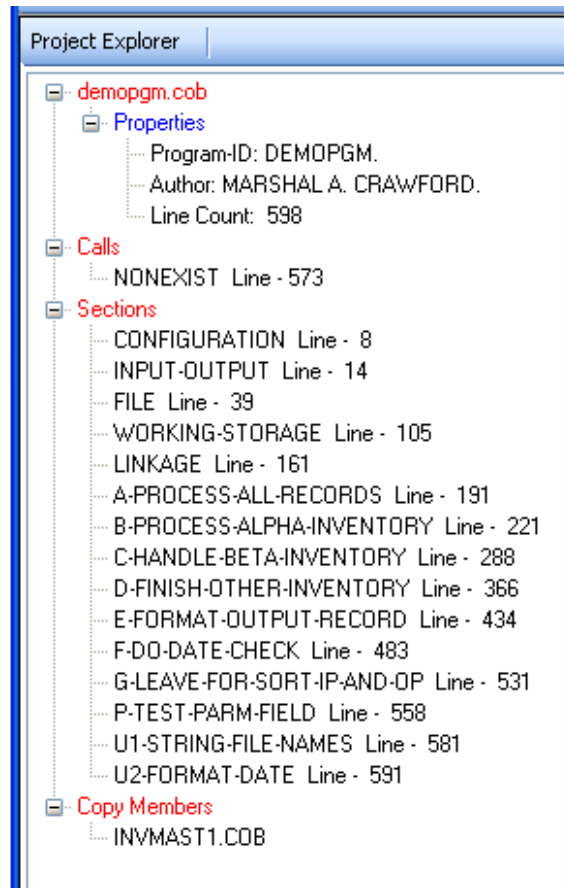
This button ONLY applies to the first tab, when multiple tabs are present. When used it saves the contents of the first tab and also reruns CAT analysis on that program.

Blue arrow button

The blue arrow button to its right refreshes the errors window, clearing corrected errors and adding newly detected ones.

Project explorer window

Other than the source code window, the project explorer window is probably the most noticeable component of the Syntax Editor. It resides on the left side of the editor. The project explorer window is another of CAT's useful features that facilitates the user's simultaneous analysis and editing of source code. It lists a program's properties, for quick reference, and lists many of the most important elements in the structure of a program. It contains the PROGRAM-ID and the AUTHOR, as well as the number of lines in the program. The listing of major program elements helps to give a bird's-eye view of the program, and gives the user quick, one-click access to these elements.

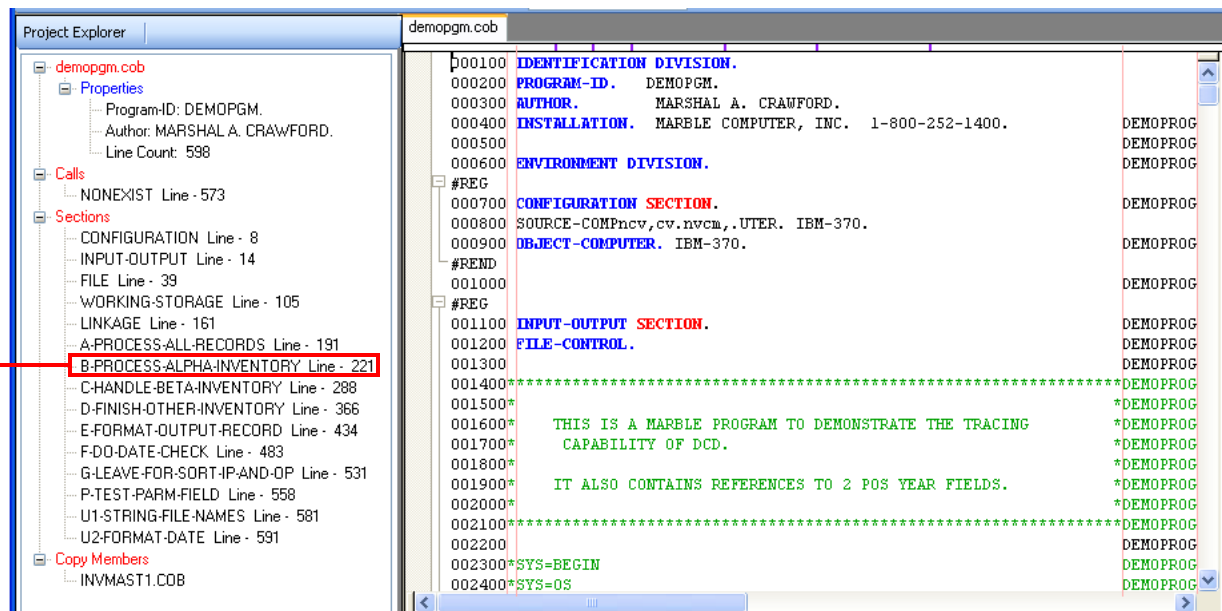


Project Explorer Window

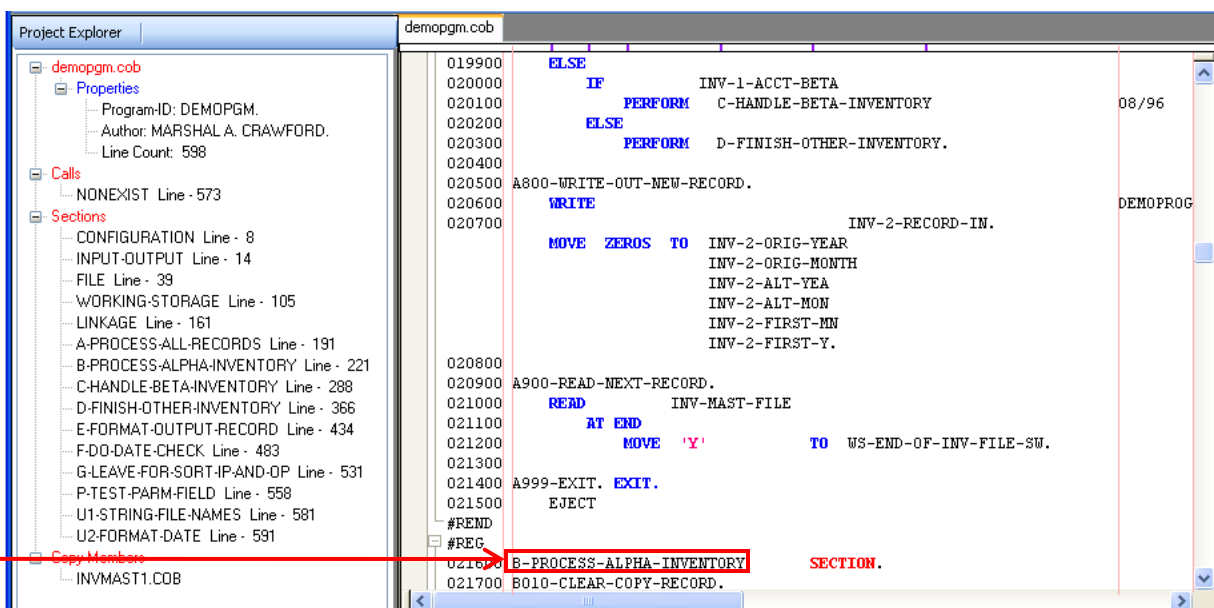
This window currently lists three types of program elements.

1. It shows the program's CALLS that are called from the program currently loaded in the editor.
2. It lists all of the SECTIONS in the program.
3. It lists each of the COPY members used in the program.

Clicking on any of these individual elements will take the source code window to that element.



Clicking on B-PROCESS-ALPHA-INVENTORY jumps to the window below



Errors Window

The errors window is located underneath the project explorer and source code windows, and above the status bar. This window is used to list coding errors detected by CAT. Each error listing is displayed in a single line of the window. Each listing includes the following information, divided into columns:

- An error code and a description of the error (Description)
- The COPY name if it is in a COPY member (Filename)
- The line the error was found in (Line)
- The column the error was found in (Column)

Additional information on the error – an explanation of the error, and action recommended to correct it – can be viewed by placing the mouse cursor over the error description. Double-clicking on one of these errors will take the source code window to that error, saving time in fixing errors.

Description	Filename	Line	Column
DCDM5J01-W - THE NAME (INV-M-3RD-DAT) DOES NOT MATCH TO A DATA DIVISION NAME		000255	1
			1
			1
			1

Error Window

Description	Filename	Line	Column
DCDM5J01-W - THE NAME (INV-M-3RD-DAT) DOES NOT MATCH TO A DATA DIVISION NAME		000255	1
DCDM5J01-W THE NAME (reference-name) DOES NOT MATCH TO A DATA DIVISION NAME Explanation: A Procedure Division reference was made to a Data Division name that does not exist. User Action: Check for a spelling error. If the name shown is a special register, notify MARBLE Computer, Inc. Then, ignore the message.			

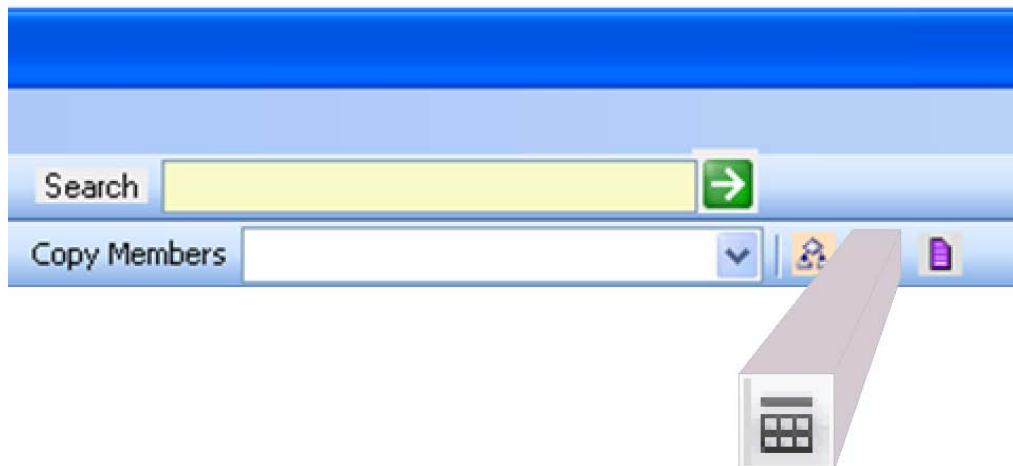
Pop-up Error Information

Note – Our Cobol errors are similar but separate from those produced by a COBOL compiler. Therefore in some instances a Cobol compiler will list an error that CAT did not catch.

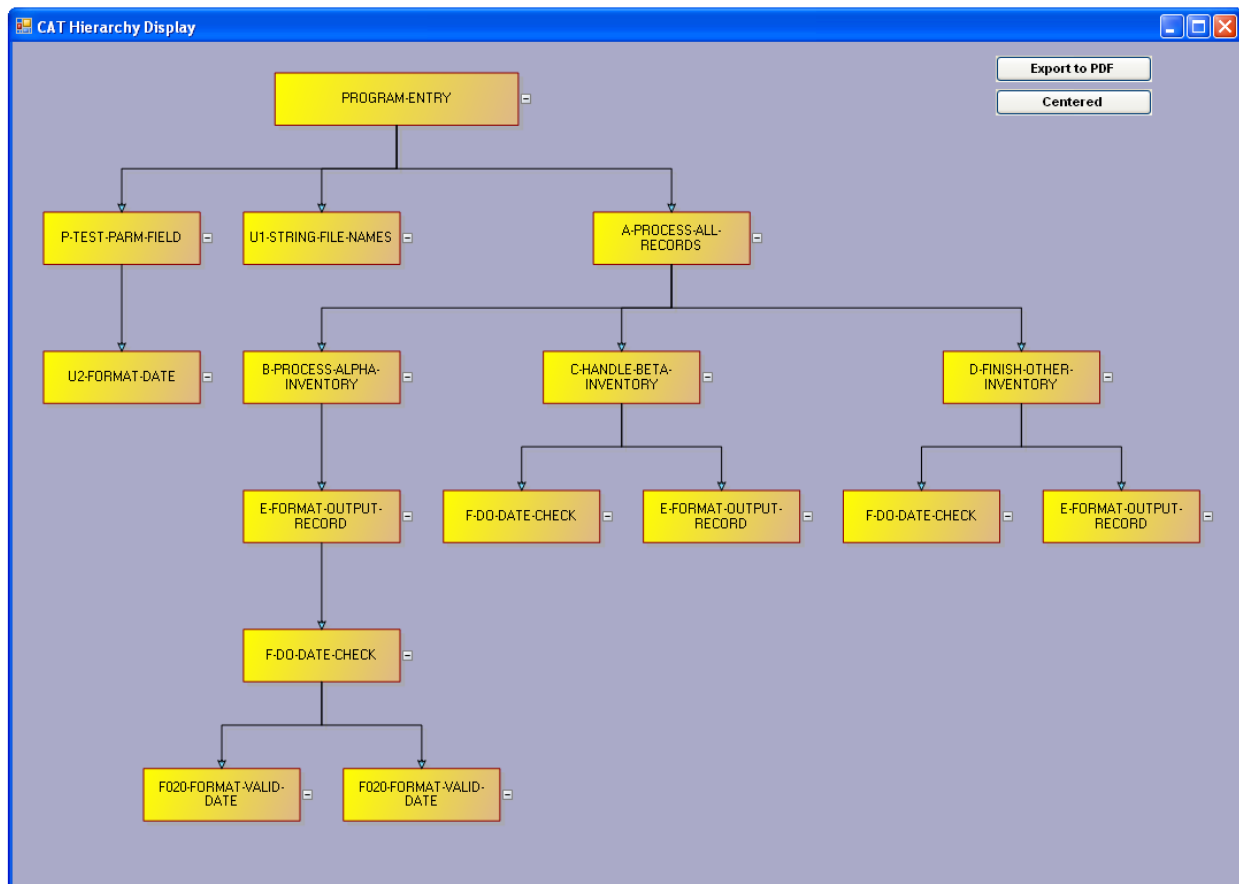
Hierarchy Charts

As a comprehensive code analysis tool, CAT includes advanced charting features. One of these is a hierarchy chart system that graphically shows the PERFORM structure of a program.

To generate a hierarchy chart of the program currently loaded in CAT, click the 'Hierarchy Chart' button just to the right of the 'FlowChart' button in CAT. See example below.



The Hierarchy Chart is a visual representation of how COBOL routines are performed.

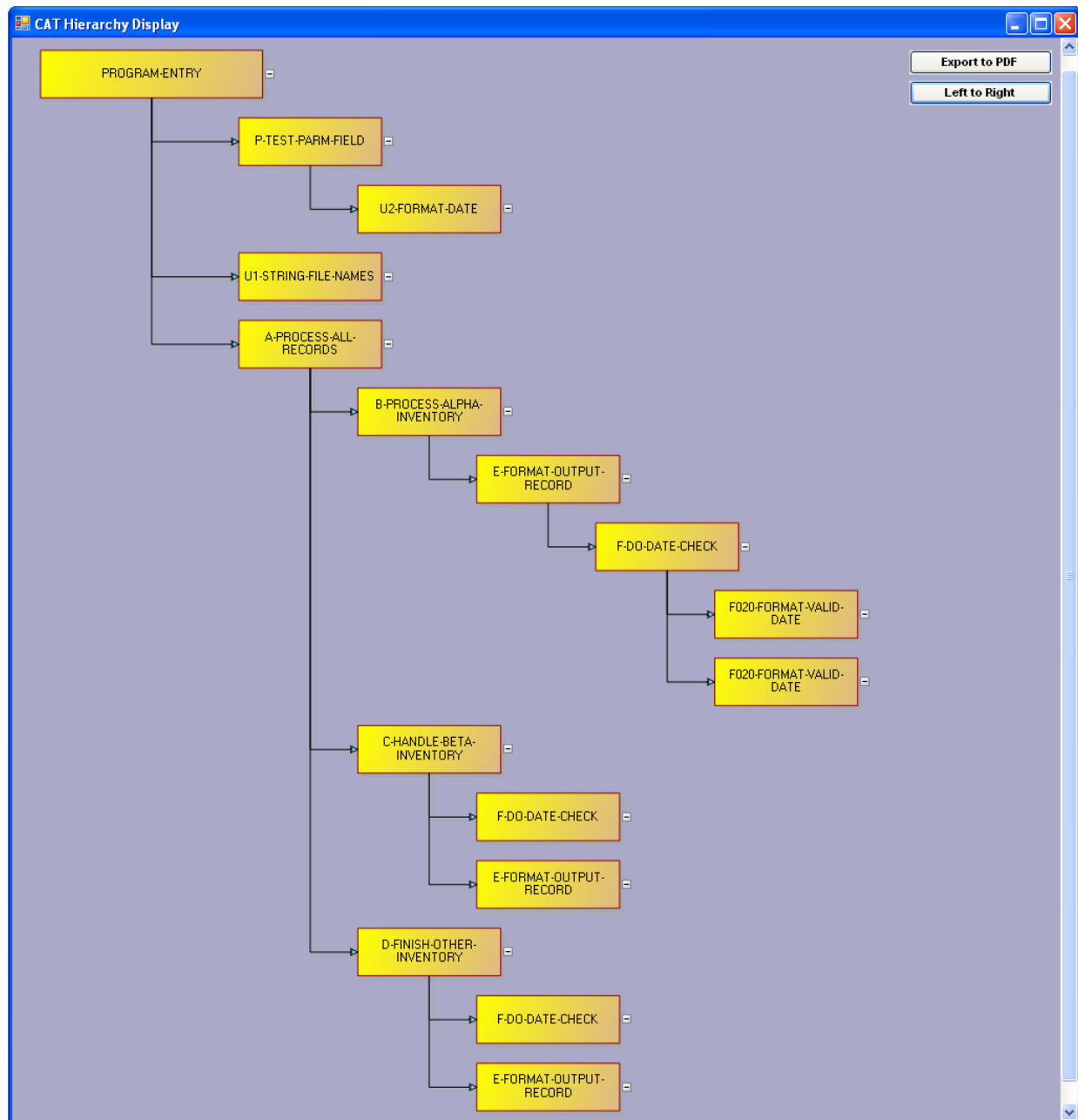


Hierarchy Chart of PERFORMed Subroutines

To start the hierarchy chart at other than PROGRAM-ENTRY, close the hierarchy chart window, point the forward tracing window within CAT to the desired starting routine, then re-open the hierarchy chart window.

The Hierarchy Chart can be saved as a PDF file by clicking the 'Export to PDF' button in the upper right-hand corner of the display.

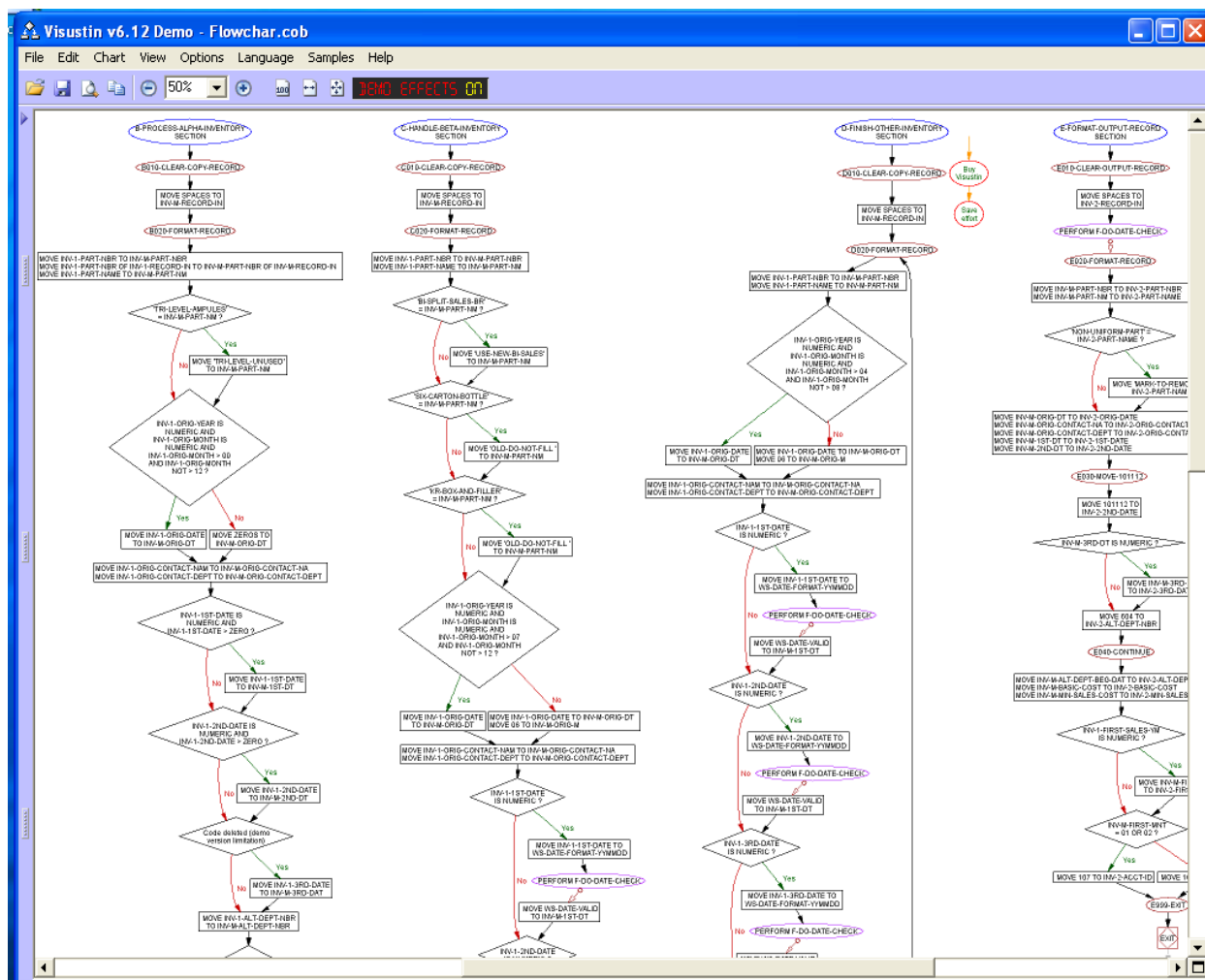
The hierarchy chart is available in two viewing layouts – top-to-bottom, shown above, and left-to-right, shown next. To change from one layout to another, click the button immediately underneath the 'Export to PDF' button in the upper right-hand corner to toggle back and forth.



Hierarchy Chart of PERFORMed Subroutines, Left-to-Right View

Flowcharts

CAT can also provide a detailed symbolic representation of programs' logic and mechanisms with flowcharts.



Detailed Flowchart Showing Individual Paragraphs, Commands and Other Elements

Special Considerations

Incorrectly Defined Mainline Routine - Caused by missing Stop Run type verb

Stop Run type verbs includes STOP RUN, GOBACK and EXIT PROGRAM.

Incorrectly defined mainline routines can cause a problem in the Quick View Window by including PERFORMs in the mainline routine that do not belong there. This can happen because the End Range address for the mainline routine is incorrect.

The end range of a mainline routine is normally determined by the presence of an **Unconditional Stop Run type Verb** (i.e. unconditional STOP RUN, GOBACK, or EXIT PROGRAM). For instance, the statement:

```
EXIT PROGRAM.
```

defined at the end of a mainline routine, before the next starting PERFORMed paragraph or section, will provide a correct end Range address to properly define the end range of the mainline routine, as long as there is not a previous unconditional Stop Run type verb ahead of it.

Conditional STOP RUN type verbs (e.g. within an IF) are not counted for creating the end range of the mainline routine. For instance, the following will not define the end of a mainline routine:

```
IF    WS-INVALID-CONDITION,  
      GOBACK.
```

There are various ways where there might NOT be a valid end program verb like STOP RUN at the end of a mainline routine. A couple of possible ways are listed below:

1. There is a CALL at the end of the mainline routine that one way or another invokes program end without actually using a STOP RUN or EXIT PROGRAM type verb.
2. There is a PERFORM at the end of the mainline routine that has a STOP RUN type verb or other embedded logic within it to end the run, therefore there is not STOP RUN type verb at the end of the mainline.

When CAT determines that the mainline routine might not be correctly determined, then one of two warning messages appear under the 'Cobol Errors' tab that indicate the situation as follows:

1. A Stop Run (or Goback or Exit Program) was NOT found within the program.

2. A Stop Run (or Goback or Exit Program) was found within the program, but it apparently is not at the end of a mainline routine.

- The algorithm used by CAT to determine a badly placed unconditional Stop Run type verb is as follows:
- The mainline routine is first set up from Program Begin (see definition for Program Begin at the end of this section) to the first found Unconditional Stop Run type verb.
- All program PERFORMs are set up with Begin and End Ranges.
- Each Perform Begin Range is examined to see if there are any Begin Ranges present in the just established Mainline Routine, meaning that for this program there is a PERFORM into the (possibly incorrectly) defined Mainline Routine. See example below:

```
PROCEDURE DIVISION.  
PARAGRAPH-1.  
    PERFORM    A-PARAGRAPH UNTIL END-JOB.  
    PERFORM    Z-STOP-RUN.  
A-PARAGRAPH.  
    MOVE FIELD-A TO FIELD-B.  
Z-STOP-RUN.  
STOP RUN.
```

- This incorrectly defined CAT mainline routine includes A-PARAGRAPH and Z-STOP-RUN within it, where ideally the mainline should end after the PERFORM Z-STOP-RUN statement.
- When this is found, CAT flags a warning suggesting that the Mainline Routine might be incorrectly defined. (While it might be possible for a technician to PERFORM a paragraph within the mainline routine, it is more unusual than usual.) Once the warning is made, it is up to the user to determine any needed correction. See below for ways how either user, or CAT with user option, will correct this situation.

Both of these two conditions may be permanently corrected with one simple surefire method which, when used, will correct either warning. This method is as follows:

- Go into the source code and determine the correct end of the mainline routine and insert a comment (* in column 7) with one of the following formats:

```
*CAT STOP RUN
```

```
*CAT GOBACK
*CAT EXIT PROGRAM
```

Leave exactly one SPACE after CAT before the next operand and if STOP RUN or EXIT PROGRAM is used, leave only one SPACE between those two operands.

Leave **no** SPACE between the asterisk and operand CAT

- When used the warning which was issued previously will disappear
- The mainline routine will now be correctly defined to CAT

How to correct these warnings without modifying the source code:

- If the warning shown indicates a missing STOP RUN type verb, CAT will make an assumption to a correct end of the mainline routine. This is done by creating the end range of the mainline routine just before the first PERFORMed paragraph or section. (See A-PARAGRAPH in margin A of above example.) The warning message still persists and it is up to the user to verify the mainline of the Quick View Window is now correct. This is done by verifying that all PERFORMs indented one level underneath the mainline routine are correct.
- If the warning shown indicates the specified STOP RUN type verb may be incorrect, the user may use the 'Narrative Selection' Tab in the TABS provided for the Quick View Window and then check the 'Create CAT Stop Run' option under the 'Special Options' heading. This option will stay in effect temporarily as long as CAT is executing and as long as this program is being analyzed. If a different program is analyzed or CAT is ended, the temporary option will be permanently turned off until it is turned back on again. The logic used here to create a new mainline End Range is the same as shown above for a missing STOP RUN type verb and the user must do appropriate verifying when using this method to insure that all PERFORMs indented one level underneath the mainline routine are now correct.

Definition of Program Begin

- Program Begin normally starts after the PROCEDURE DIVISION statement. However, if a DECLARATIVES SECTION is present, the PROCEDURE DIVISION begins just after the end of that SECTION.